

V4 6.018, 526

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-4247

(43) 公開日 平成11年(1999) 1月6日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

H 0 4 L 12/44  
12/46  
12/28  
12/56

H 0 4 L 11/00  
11/20

3 4 0  
3 1 0 C  
1 0 2 D

審査請求 未請求 請求項の数21 O L 外国語出願 (全 101 頁)

(21) 出願番号 特願平10-80117

(22) 出願日 平成10年(1998) 2月20日

(31) 優先権主張番号 08/803116

(32) 優先日 1997年2月20日

(33) 優先権主張国 米国 (U S)

(71) 出願人 598040411

マクロニクス アメリカ インコーポレイ  
テッド

アメリカ合衆国 カリフォルニア州  
95131 サン ホセ リダー パーク ド  
ライヴ 1338

(72) 発明者 チャン チー リウ

アメリカ合衆国 カリフォルニア州  
94539 フリーモント ボーニー ドライ  
ヴ 44948

(74) 代理人 弁理士 中村 稔 (外7名)

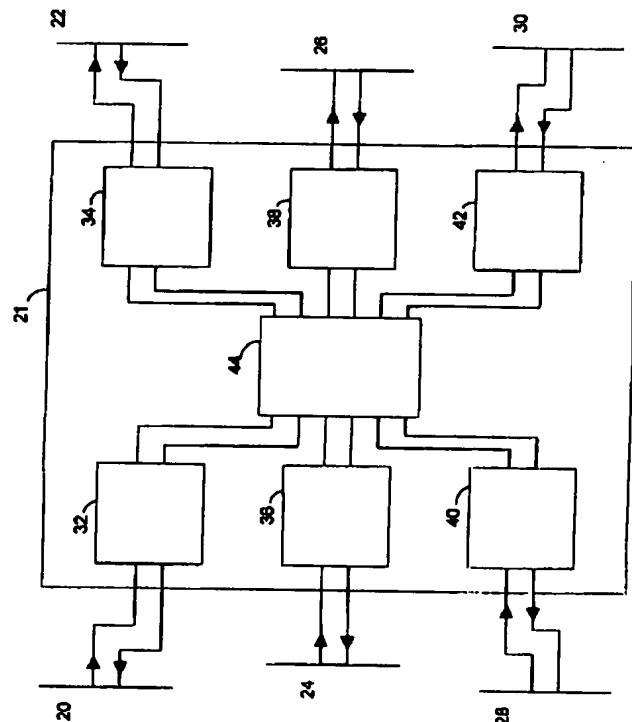
最終頁に続く

(54) 【発明の名称】 ネットワークメディアと集積回路間の自己学習機能を有するブリッジデバイス及びこれに基づく方法

(57) 【要約】 (修正有)

【課題】 ブリッジにおいて、複雑性とコストの減少、及びローカルパケットが他のネットワークセグメントへリークすることを防止する。

【解決手段】 ブリッジデバイス21は、種々のポート32～42に受け入れられたパケットのソースアドレスを観察することによって、ネットワークセグメント20～30がアドレスに関係していることを学習する。ブリッジデバイスが新しいパケットを受けると、パケットのソースアドレスに関するアドレスの位置の知識を更新し、またパケットの宛先アドレスに基づいたパケットの転送の要否を決める。フィルタ44が、パケットをあるネットワークセグメントから別のネットワークセグメントに選択的にブロックし、又は渡す。



## 【特許請求の範囲】

【請求項 1】 第 1 のネットワーク媒体に接続される第 1 のポートと、第 2 のネットワーク媒体に接続される第 2 のポートと、アドレスのそれぞれのセットに対応し、アドレスの前記それぞれのセットの少なくとも 1 つのアドレスが第 1 のネットワーク媒体を介してアクセスできるかどうかを指示する、第 1 の複数の指示と、アドレスのそれぞれのセットに対応し、アドレスの前記それぞれのセットの少なくとも 1 つのアドレスが第 2 のネットワーク媒体を介してアクセスできるかどうかを指示する、第 2 の複数の指示と、をストアするメモリと、前記第 1 のポートと、前記第 2 のポートと、前記メモリとに接続されて、前記ポートが宛先アドレスを有するバケットを前記第 1 のネットワークから前記第 2 のネットワークにパスするか又はブロックするようにさせる接続回路とを含む第 1 のネットワーク媒体を第 2 のネットワーク媒体に接続するためのデバイスであって、前記接続回路は、前記ポートが、前記第 1 の複数の指示からの、前記バケットの前記宛先アドレスを含むアドレスのセットに対応する第 1 の指示と、第 2 の複数の指示からの、前記バケットの前記宛先アドレスを含む前記アドレスのセットに対応する第 2 の指示とに基づいてパス又はブロックさせるようにするものであるデバイス。

【請求項 2】 バケットのソースアドレスを読み込む第 1 の回路を更に有し、前記第 1 の回路は、バケットのソースアドレスに対応する第 3 の指示を、第 1 の複数の指示に設定する、請求項 1 に記載のデバイス。

【請求項 3】 第 2 のポートからの第 2 のバケットのソースアドレスを読み込む第 2 の回路を更に有し、前記第 2 の回路は複数の第 2 の指示に、第 2 のバケットのソースアドレスに対応する第 4 の指示を設定するものである、

請求項 2 に記載のデバイス。

【請求項 4】 中央演算処理装置（CPU）が指示をメモリに設定できるようにする、メモリに接続された CPU インターフェースを更に有する、請求項 2 に記載のデバイス。

【請求項 5】 古いエントリをメモリから消去する、メモリに接続された消去回路を更に有する、請求項 2 に記載のデバイス。

【請求項 6】 エージングタイマと、前記エージングタイマ及び前記メモリに接続された消去回路と、を更に含み、前記第 1 の複数の指示は、第 3 の複数の指示と、第 4 の複数の指示と、を含み、

前記第 1 の回路は、第 3 の指示を前記エージングタイマに基づいて第 3 の複数の指示又は第 4 の複数の指示に設定するように構成され、前記消去回路は、前記エージングタイマに基づいて、第 3 の複数の指示又は第 4 の複数の指示を消去するように構成される、請求項 2 に記載の

デバイス。

【請求項 7】 エージングタイマと、前記エージングタイマ及び前記メモリに接続された消去回路と、を更に有し、

前記第 1 の複数の指示は、

第 3 の複数の指示と、

第 4 の複数の指示と、を含み、

前記第 2 の複数の指示は、

第 5 の複数の指示と、

10 第 6 の複数の指示と、を含み、

前記第 1 の回路は、第 3 の指示を前記エージングタイマに基づいて、第 3 の複数の指示又は第 4 の複数の指示に設定するように構成され、

前記第 2 の回路は、第 4 の指示を前記エージングタイマに基づいて、第 5 の複数の指示又は第 6 の複数の指示に設定するように構成され、

前記消去回路は、前記エージングタイマに基づいて第 3 の複数の指示又は第 4 の複数の指示を消去するように構成され、前記消去回路は、エージングタイマに基づいて第 5 の複数の指示又は第 6 の複数の指示を消去するように構成される、請求項 3 に記載のデバイス。

【請求項 8】 第 1 の指示が、第 2 のセットのアドレスのうちの少なくとも 1 つのアドレスが第 1 のネットワーク媒体を介してアクセス可能でありうることを指示し、第 2 の指示が、第 4 のセットのアドレスのうちの少なくとも 1 つのアドレスが、第 2 のネットワーク媒体を介してアクセス可能であり得ることを指示しない、ならば、接続回路は、第 1 のネットワークセグメントから第 2 のネットワークセグメントへのバケットをブロックするために構成されている、請求項 1 に記載のデバイス。

【請求項 9】 第 2 の指示が、バケットの宛先アドレスが第 2 のネットワークセグメントを介してアクセス可能でありうることを指示するならば、接続回路は、バケットを第 1 のネットワークセグメントから第 2 のネットワークセグメントに通すように構成されている、請求項 1 に記載のデバイス。

【請求項 10】 バケットの宛先アドレスに応じて第 1 のインデックスを生成するインデックス生成回路と、前記第 1 のインデックスに基づく第 1 の複数の指示から第 1 の指示を選択する選択回路と、を更に有し、前記選択回路が、接続回路及びインデックス生成回路に接続される、請求項 1 に記載のデバイス。

【請求項 11】 インデックス生成回路がハッシュ回路を含み、第 1 のインデックスがバケットの宛先アドレスのハッシュ関数の結果を含む、請求項 10 に記載のデバイス。

【請求項 12】 ハッシュ回路が、周期冗長検査（CRC）回路を含む、請求項 11 に記載のデバイス。

【請求項 13】 アドレスのそれぞれのセットが、媒体アクセス制御（MAC）アドレスを含む、請求項 1 に記

載のデバイス。

【請求項 1 4】 第 1 のネットワーク媒体が、キャリア検出多重アクセスプロトコルを使用するネットワークを含む、請求項 1 に記載のデバイス。

【請求項 1 5】 第 1 のネットワーク媒体が、衝突ドメインネットワークセグメントを有する、請求項 1 に記載のデバイス。

【請求項 1 6】 全てのデバイスが、単一の集積回路に装備されている、請求項 1 に記載のデバイス。

【請求項 1 7】 パケットを第 1 のネットワーク媒体から第 2 のネットワーク媒体に選択的に通す方法であって、前記方法が、

第 1 の指示及び第 2 の指示に基づいて第 1 のパケットを第 1 のネットワークセグメントから第 2 のネットワークセグメントへ通すか又はブロックすることを、含み、前記第 1 の指示が、第 1 のパケットの宛先アドレスに対応し、前記宛先アドレスを含むアドレスの第 1 のセットのアドレスが第 1 のネットワーク媒体を介してアクセス可能でありうるかどうかを指示し、

前記第 2 の指示が、第 1 のパケットの宛先アドレスに対応し、前記宛先アドレスを含むアドレスの第 2 のセットのアドレスが第 2 のネットワーク媒体を介してアクセス可能でありうるかどうかを指示する、前記方法。

【請求項 1 8】 第 1 の指示及び第 2 の指示に基づいて、第 1 のパケットを第 1 のネットワークセグメントから通すか又はブロックするステップが、第 2 の指示が、アドレスの第 2 のセットにおけるアドレスが第 2 のネットワーク媒体を介してアクセス可能であり得ることを指示するならば、パケットを第 1 のネットワークセグメントから第 2 のネットワークセグメントに通すことを、更に有する、請求項 1 7 に記載の方法。

【請求項 1 9】 第 1 の指示及び第 2 の指示に基づいて、第 1 のネットワークセグメントからの第 1 のパケットを通すか又はブロックするステップが、第 1 の指示が、アドレスの第 1 のセットのアドレスが第 1 のネットワーク媒体を介してアクセス可能であり得ることを指示し、

第 2 の指示が、アドレスの第 2 のセットのアドレスが第 2 のネットワーク媒体を介してアクセス可能であり得ることを指示しないならば、第 1 のネットワークセグメントから第 2 のネットワークセグメントへ第 1 のパケットをブロックする、

ならば、第 1 のパケットを第 1 のネットワークセグメントから第 2 のネットワークセグメントへブロックすることを更に含む、請求項 1 7 に記載の方法。

【請求項 2 0】 第 1 のハッシュ値を生成するために、第 1 のネットワークセグメントのノードのアドレスでハッシュ関数を実行し、第 1 のハッシュ値によって第 1 の指示を指標付けする、ステップを更に含む、請求項 1 7 に記載の方法。

【請求項 2 1】 ハッシュ関数を実行するステップが、第 1 のネットワークセグメントのノードのアドレスで周期冗長検査を実行することを含む、請求項 2 0 に記載の方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】本発明は、相互接続ネットワーク用デバイスの分野に関し、特に、相互接続ネットワークに関する自己学習機能を有するブリッジデバイスに関する。

【0 0 0 2】

【従来の技術】ローカルエリアネットワーク（LAN）がコンピュータ適用業務で使用される状況が増加している。LAN プロトコルの 1 つのタイプがイーサネットプロトコルである。イーサネットプロトコルでは、ネットワークは種々のセグメント即ち「衝突領域」に分けられている。各ネットワークセグメントは複数のノードからなっている。ノードは、ターミナル又はプリンタのようなネットワークのデバイスである。セグメントのノードが他のノードから突き止められ得る距離は、限られている。例えば、セグメント内の「ファーストイーサネット」では、ノードは 2 0 5 m まで離すことができるだけである。2 つのセグメントは、ブリッジデバイスを介して接続することができる。ブリッジは、1 つのセグメント内で生ずる衝突が別のセグメントに影響しないように衝突領域を隔離する。一旦、2 つのセグメントの衝突領域がブリッジによって分けられたならば、2 つのノードは、同じ衝突領域にあったとしても、前にあった状態より遠くに離れたものとすることができる。

【0 0 0 3】セグメンテーションの他の別のブリッジ機能は、フィルタリングである。フィルタリングの目的は、別のセグメントから入り込み、バンド幅を無駄にする好ましくない情報のトラフィックを遮断することである。ブリッジは、情報をパケットの形であるセグメントから別のセグメントへ渡す。種々の基準に基づいて、ブリッジは、いくつかのパケットを、それを受け取る場所に転送しない。例えば、ブリッジと、テーブルのデータにアクセスするための方法と、これを遠隔局間でのデータのルーティングのために適用することは、1 9 9 0 年 4 月 2 5 日に発行された欧州特許公開番号 0, 3 6 5, 3 3 7, A 2（出願番号 8 9 3 1 0 7 8 9, 6）の Marshall による特許出願に記載されている。例えば、LAN 間接続装置は、米国特許第 5, 4 7 7, 5 4 7 号（Sugiyama）に記載されている。

【0 0 0 4】一つの基準として、パケットが同じセグメント（ローカルトラフィック）内の別のノードに向けられている場合、ブリッジはパケットを別のセグメントに転送すべきではないとすることが好ましい。このために、ブリッジは、特別なノードが特別なセグメントに存在するかどうかに関する情報を必要とする。ブリッジが

5

この情報を得ることができる一つの方法は、各セグメントからのパケットのソースアドレスを観察し、ソースアドレステーブルのあるセグメントからのパケットからソースアドレスをストアすることによる。パケットが到着するとき、CPUは、多くのエントリのテーブルを探索し、行先アドレスをそれらの各々と比較しなければならない。変形実施形態としてCPUを使用して、探索及び比較機能はまた、連想記憶装置(CAM)で実施される。CAMを使用することは、余分なハードウェア(CAM)を要求し、ギガバイト/秒の速度を有するネットワークを基準化することは困難若しくは不可能である。FDDIブリッジフレーム学習と、フィルタリング装置と、ソースアドレスがCAMにストアされている方法とが、米国特許第5,481,540号(Gang)に記載されている。

【0005】従って、パケットが別のネットワークセグメントに転送されるべきであるかどうか判断するために、ブリッジの効率的で低コストの探索が必要である。ブリッジがテーブルの情報をストアすることによってノードの位置を学習するならば、次いで、ノードがその後移動されるならば、テーブルはもはや正確でなくても良い。それゆえ、テーブルがノードの位置のより最近のビューを反映するように、ブリッジのテーブルの更新をするための装置及び方法の必要性がある。ソースアドレスに関する情報を有するテーブルが、ハッシュ関数によって指標付けられ得る。ハッシュ関数は、エントリをアドレスに関するテーブルに指標付けするために用いられ得る。しかしながら、ハッシュ関数の問題は、2つのアドレスが同じハッシュ値にマップし得ることである。例えば、ハッシングが使用される相互接続ネットワークのためのアドレス検査回路を備えるブリッジ装置は、米国特許第5,247,620号(Fukuzawa)に記載されている。

【0006】従って、ブリッジに関する複雑性及びコストを低減させるのを助け、ネットワーク媒体間の相互接続の機能性をより効率的に提供するのを助け、他のネットワークセグメントへのローカルパケットのリークを避けるのを助け、高速ネットワークのためにスケーラブルである、デバイスの必要性がある。

【0007】

【課題を解決するための手段】本発明は、第1のネットワーク媒体を第2のネットワーク媒体に接続するためのデバイスを提供する。第1のポートは第1のネットワーク媒体に接続され、第2のポートは第2のネットワーク媒体に接続される。メモリが、第1の複数の指示と第2の複数の指示とをストアする。第1の複数の指示の指示が、それぞれのセットのアドレスに対応し、それぞれのセットのアドレスの少なくとも1つのアドレスが第1のネットワーク媒体を介してアクセス可能であり得るかどうかを指示する。第2の複数の指示の指示が、それぞれ

(4)

特開平11-4247

6

のセットのアドレスに対応し、それぞれのセットのアドレスの少なくとも1つのアドレスが第2のネットワーク媒体を介してアクセス可能であり得るかどうかを指示する。接続回路が、第1のポート、第2のポート及びメモリに接続されている。接続回路によって、ポートは第1のネットワークから第2のネットワークにパケットを通させるか又はブロックさせることができる。パケットは宛先アドレスを有する。接続回路によって、ポートは、第1の複数の指示からの第1の指示と第2の複数の指示からの第2の指示とに基づいて通し又はブロックすることができる。第1の指示は、パケットの宛先アドレスを含むアドレスのセットに対応する。第2の指示は、パケットの宛先アドレスを含むアドレスのセットに対応する。

【0008】本発明の実施形態は、パケットのソースアドレスを読み込む第1の回路を含む。第1の回路は、第3の指示をパケットのソースアドレスに対応する第1の複数の指示に設定する。第2の回路は、第2のパケットのソースアドレスを第2のポートから読み込む。第2の回路は、第4の指示を第2の複数の指示に設定する。第4の指示は、第2のパケットのソースアドレスに対応する。本発明の実施形態は、中央演算処理装置(CPU)が指示をメモリに設定することができる、メモリに対応するCPUインターフェースを含む。本発明の実施形態は、古いエントリをメモリから消去する、メモリに対応する消去回路を含む。本発明の実施形態は、エージングタイマ及びメモリに対応するエージングタイマ及び消去回路を含む。本発明のこの実施形態では、第1の複数の指示は、第3の複数の指示と第4の複数の指示とを含む。第1の回路は、第3の指示をエージングタイマに基づいて第3の複数の指示及び第4の複数の指示に設定するように構成されている。消去回路は、エージングタイマに基づいて第3の複数の指示及び第4の複数の指示を消去するように構成されている。本発明の実施形態では、第1の指示が、第2のアドレスのセットの少なくとも1つのアドレスが第1のネットワーク媒体を介してアクセス可能であり得ることを示し、第2の指示が、第4のアドレスのセットの少なくとも1つのアドレスが第2のネットワーク媒体を介してアクセス可能であり得ることを指示しない、ならば、接続回路は、パケットを第1のネットワークセグメントから第2のネットワークセグメントにブロックするように構成される。

【0009】本発明の変形実施形態では、第2の指示が、パケットの宛先アドレスが第2のネットワークセグメントを介してアクセス可能であり得ることを指示するならば、接続回路は、第1のネットワークセグメントから第2のネットワークセグメントにパケットを通すように構成される。本発明の変形実施形態は、パケットの宛先アドレスに対応する第1のインデックスを生成するインデックス生成回路と、第1のインデックスに基づいて

第1の複数の指示から第1の指示を選択するセレクト回路とを含む。セレクト回路は、接続回路とインデックス生成回路とに接続される。本発明の変形実施形態では、インデックス生成回路はハッシュ回路を含み、第1のインデックスはパケットの宛先アドレスのハッシュ関数の結果を含む。

【0010】本発明の変形実施形態では、アドレスのそれぞれのセットが、媒体アクセス制御（MAC）アドレスを含む。本発明の変形実施形態では、第1のネットワーク媒体は、キャリア検出多重アクセスプロトコルを使用するネットワークを含む。本発明の別の実施形態では、第1のネットワーク媒体は衝突ドメインネットワークセグメントを含む。変形実施形態は、第1のネットワーク媒体を第2のネットワーク媒体に接続するための媒体アクセス制御（MAC）層デバイスである。第1のポートは、第1のネットワーク媒体に接続される。第2のポートは、第2のネットワーク媒体に接続される。メモリは、第1のテーブル及び第2のテーブルを有する。第1の回路は、第1のポートからの第1のパケットを監視し、エントリを第1のパケットに対応して第1のテーブルに設定する。第1のテーブルのエントリは、第1のポートからの第1のパケットのソースアドレスに基づいて指標付けされる。第1のテーブルの少なくとも1つのエントリは、1以上のアドレスにマップされる。第2の回路は、第2のポートからの第2のパケットを監視し、エントリを第2のパケットに対応して第2のテーブルに設定する。第2のテーブルのエントリは、第2のポートからの第2のパケットのソースアドレスに基づいて指標付けされる。第3の回路は、第1のポートからの第1のパケットの宛先アドレスに基づいて、第1のテーブルからの第1のエントリ及び第2のテーブルからの第2のエントリを選択する。第1のポートからの第1のパケットのソースアドレスは、MACアドレスを有し、第2のポートからの第2のパケットのソースアドレスは、MACアドレスを有する。第3の回路は、第1のエントリが設定されておらず、若しくは、第1のエントリ及び第2のエントリの両方が設定されているならば、第1のパケットを第1のポートから第2のポートに通す。

【0011】本発明は、ノード間の距離が衝突ドメイン制限を越えて拡張することができるように、ファースト・イーサネット・リピータを互いにリンクさせるために経済的な解を与えるのを助ける。ブリッジによって結合された衝突ドメインネットワークを設計することによって、柔軟性が達成される。自己学習は、デバイスをプログラミングする必要を取り除くのを助ける。学習及びフィルタリング方法は、専用CAMハードウェアの必要性を取り除くのを助ける。本発明の他の態様及び利点は、図面、詳細な説明及び特許請求の範囲から理解することができる。

【0012】

【発明の実施の形態】本発明の好ましい実施形態の詳細な説明を、図を参照して行う。図1は、本発明の実施形態のアーキテクチャと、本発明の使用に関するコンテキストを図示する。セグメント20、24、26、28及び30は、ブリッジデバイス21を介して相互接続されている。ブリッジデバイス21は、セグメント20、24、26、28及び30から情報を受け、かかる情報を他のセグメントに渡すかどうか判断する。パケットはローカルアドレス（パケットが生成されたセグメント内部）、又は、非ローカルアドレス（パケットが生成されたセグメントの外部）に送られる。ブリッジ21は、ローカルアドレスに宛先付けされたパケットをブロックし、非ローカルアドレスに宛先付けされたパケットを渡すのを助ける。フィルタ44は、パケットをあるネットワークセグメントから別のネットワークセグメントに選択的にブロックし、または、渡す。

【0013】種々のポート32、34、36、38、40及び42に受け入れられたパケットのソースアドレスを観察することによって、ブリッジデバイス21は、ネットワークセグメントがアドレスに関係していることを学習する。ブリッジデバイス21が新しいパケットを受け取るとき、ブリッジデバイス21は、パケットのソースアドレスに関するアドレスの位置のその知識を更新し、また、パケットの宛先アドレスに基づいたパケットを転送するかどうか決定する。図1は、合計6つのネットワークセグメント（20、24、28、22、26及び30）を相互接続するブリッジを示す。変形実施形態のブリッジ21は、別の数のネットワークセグメントを相互接続するために実行され得る。例えば、ブリッジ21は、合計2つのネットワークセグメントを相互接続するように設計され得る。ある好ましい実施形態では、ブリッジ21は、2つのネットワークセグメントに関する制御回路を含む単純な集積回路を含む。図2は、本発明に関するパケットを渡すためのブロックを示すブリッジデバイスの概略ブロック図である。図2は、ネットワークノード46、48、50及び52を含むネットワークセグメントA53と、ネットワークノード54、56、58、60及び62を含む第2のネットワークセグメントB55と、ネットワークセグメント53と55を相互接続するブリッジデバイス45とを含む。ブリッジデバイス45は、フィルタ64、及び、動的テーブルA066と、動的テーブルA168と、静的テーブルA70と、動的テーブルB072と、動的テーブルB174と、静的テーブルB76とを有するメモリ65を含む。フィルタ64は、動的テーブルA066と、動的テーブルA168と、静的テーブルA70と、動的テーブルB072と、動的テーブルB174と、静的テーブルB76に含まれる情報に基づいて、パケットをネットワークセグメントA53からネットワークセグメントB55に渡すかどうか判断する。テーブ

ルは、ノードがブリッジの特別なサイドに存在するかどうかの指示を含む。指示がハッシュ値に基づいて指標付けされているので、2つの異なるアドレスが同じハッシュ値にマップされ得るという可能性がある。この可能性のために、正の指示がネットワークセグメントAからのパケットのアドレスに対応するサイドAテーブルに見つけられるならば、フィルタはまた、サイドBテーブルをチェックする。図3は、本発明に関する学習のためのブリッジデバイス45の概略ブロック図である。図3は、ソースアドレス78と宛先アドレス80とを含むセグメントAからのパケットと、ソースアドレス82と宛先アドレス84とを含むセグメントBからのパケットとを示す。ブリッジ45は、ハッシュ回路86と、動的テーブルA0 66と、動的テーブルA1 68と、静的テーブルA 70と、ハッシュ回路88と、動的テーブルB0 72と、動的テーブルB1 74と、静的テーブルB 76とを含む。

【0014】ブリッジデバイス45がアドレスを学習するとき、パケットのソースアドレスに対応する指示は、パケットが発信されるネットワークセグメントに関するテーブルにストアされる。ソースアドレス78は、ハッシュ値を生成するためにハッシュ回路86によってハッシュされ、指示は、テーブルA0 66又はテーブルA1 68のいずれかのハッシュ値によって指標付けされてストアされる。セグメントB55からのソースアドレス82は、ハッシュ値を生成するためにハッシュ回路88によってハッシュされ、指示は、テーブルB0 72又はB1 74のいずれかにストアされ、ハッシュ値によって指標付けられる。ブリッジ45の各サイドの複数の動的テーブルの使用によって、テーブルの古いエントリをエージング及びフラッシングさせることができる。例えば、古い指示が動的テーブルA1 68にストアされている間、新しい指示を動的テーブルA0 66にストアすることができ得る。ある時間（例えば、5分）後、テーブルA1 68からの古いエントリが消去され、次いで、動的テーブルA0 68が古いテーブルになり、新しいエントリが動的テーブルA1 68に書き込まれる。同様に、動的テーブルB0 72及び動的テーブルB1 74は、指示をエージングすることができ、且つ、古い指示を消去することができるよう使用される。静的テーブルA70及び静的テーブルB76は、それらの存在が動的テーブルに記録されないように、それらのネットワークノードがそれらの存在を正常にブロードキャストしないとき、ノードがブリッジの特定のサイドに存在するかどうかの指示を提供するためにCPUによって書き込まれる。

【0015】図4は、本発明に関するブリッジデバイスのより詳細なブロック図である。図4は、セグメントA90からのソースアドレス94と、セグメントA90からの宛先アドレス96と、セグメントB164からのソ

ースアドレス128と、セグメントB164からの宛先アドレス130と、セグメントA90又はセグメントB164から生成されるパケットをフィルタリングするための支援回路とを含む。ポート92は、ネットワークセグメントA90に接続され、パケットをセグメントA90から受け取る。ポート92は、パケットをストアするためにバッファ126に接続される。ポート92は、ソースアドレス94をハッシュ回路98に提供する。ソースアドレス94、宛先アドレス96、宛先アドレス130及びソースアドレス128は、各々6バイトMACレベルアドレスである。ハッシュ回路98は、6バイトソースアドレス94に応じて8ビットハッシュ値を提供する。ハッシュ回路98は、アドレスデコーダ100に接続される。アドレスデコーダ100は、エージングスイッチ102を介して、動的テーブルA0 66と動的テーブルA1 68とに接続される。動的テーブルA0 66及び動的テーブルA1 68は、指示に対応するソースアドレスがネットワークセグメントA90に存在することを指示するために、ハッシュ回路98によって生成されるハッシュ値によって指標付けされた指示をストアする。エージングスイッチ102は、タイマ104に接続され、テーブルA0 66又はテーブルA1 68の指示がストアされるように制御する。消去回路106は、動的テーブルA0 66と動的テーブルA1 68とに接続され、タイマ104に基づいて動的テーブルの古いエントリを消去する。

【0016】ハッシュ回路108は、ポート92に接続され、セグメントA90からのパケットから宛先アドレス96を受け取り、ハッシュ値を形成するために宛先アドレスをハッシュする。ハッシュ回路108からのハッシュ値は、ハッシュテーブルに指標付けし、ハッシュ値に対応する指示を得るのに使用される。ハッシュ回路108は、アドレスデコーダ110と、アドレスデコーダ112と、アドレスデコーダ114と、アドレスデコーダ116と、アドレスデコーダ118と、アドレスデコーダ120とに接続される。アドレスデコーダ110は、動的テーブルA0 66に接続される。アドレスデコーダ112は、動的テーブルA1 68に接続される。アドレスデコーダ114は、静的テーブルA70に接続される。アドレスデコーダ116は、動的テーブルB0 72に接続される。アドレスデコーダ118は、動的テーブル174に接続される。アドレスデコーダ120は、静的テーブルB76に接続される。判断回路122の入力は、アドレスデコーダ110と、アドレスデコーダ112と、アドレスデコーダ114と、アドレスデコーダ116と、アドレスデコーダ118と、アドレスデコーダ120とに接続されている。判断回路122の出力は、スイッチ124を介して、バスの制御に接続されている。スイッチ124を介するバスの入力、バッファ126に接続される。スイッチ124を介するバ

スの出力は、セグメントB 1 6 4に接続されるポート1 6 2に接続される。

【0 0 1 7】セグメントB 1 6 4は、ポートB 1 6 2に接続される。ハッシュ回路1 3 2は、ソースアドレス1 2 8をポート1 6 2から受け取る。ハッシュ回路1 3 2は、アドレスデコーダ1 3 4に接続され、ハッシュ値をアドレスデコーダ1 3 4に提供する。アドレスデコーダ1 3 4は、アドレスデコーダ1 3 4からのハッシュ値によって指標付けされたテーブルB 0 7 2又はテーブルB 1 7 4に指示を書き込むために、エージングスイッチ1 3 6を介して動的テーブルB 0 7 2と動的テーブルB 1 7 4とに接続される。エージングスイッチ1 3 6は、タイマ1 3 8に接続され、タイマ1 3 8に依存して書き込まれるテーブル指示を制御する。消去回路1 4 0は、動的テーブルB 0 7 2と動的テーブルB 1 7 4に接続され、それぞれのテーブルの古いエントリを消去する。CPUインターフェース1 6 6は、CPUと、静的テーブルA 7 0及び静的テーブルB 7 6とに接続され、CPUインターフェースによってCPUが静的テーブル内にエントリを書き込むことができる。

【0 0 1 8】ハッシュ回路1 4 1は、宛先アドレス1 3 0を受け取り、宛先アドレス1 3 0に対応したハッシュ値を提供するためにポート1 6 2に接続される。ハッシュ回路1 4 1は、アドレスデコーダ1 4 2と、アドレスデコーダ1 4 6と、アドレスデコーダ1 4 8と、アドレスデコーダ1 5 0と、アドレスデコーダ1 5 2と、アドレスデコーダ1 5 4とに接続されている。アドレスデコーダ1 4 2は、動的テーブルB 0 7 2に接続される。アドレスデコーダ1 4 6は、動的テーブルB 1 7 4に接続される。アドレスデコーダ1 4 8は、静的テーブルB 7 6に接続される。アドレスデコーダ1 5 0は、動的テーブルA 0 6 6に接続される。アドレスデコーダ1 5 2は、動的テーブルA 1 6 8に接続される。アドレスデコーダ1 5 4は、静的テーブルA 7 0に接続される。判断回路1 5 6の入力は、アドレスデコーダ1 4 2と、アドレスデコーダ1 4 6と、アドレスデコーダ1 4 8と、アドレスデコーダ1 5 0と、アドレスデコーダ1 5 2と、アドレスデコーダ1 5 4とに接続されている。判断回路1 5 6の出力は、回路1 5 8を介してパスの制御入力に接続される。回路1 5 8を介するパスはバッファ1 6 0とポート9 2とに接続される。

【0 0 1 9】書込制御状態機械1 6 8は、ソースアドレスに対応する指示を書き込む間、ブリッジの作動を制御する。読取制御状態機械1 7 0は、指示を読み込み、ネットワークセグメントからのパケットを通し又はブロックすることを制御する。セグメントA 9 0からの新しいパケットがポート9 2に到着したとき、パケット9 4のソースアドレスは、8ビットハッシュ値を生成するためにハッシュされる。1ビット指示が、動的テーブル（動的テーブルA 0 6 6又は動的テーブルA 1 6 8）の

うちの一つにストアされる。この1ビット指示は、パケットのソースアドレスをハッシュすることによって得られるハッシュ値によって指標付けされる。各テーブルは、2 5 6の可能なハッシュ値に対応するエントリを提供するために2 5 6の1ビットエントリを含む。テーブルA 0 6 6又はテーブルA 1 6 8は、タイマ1 0 4に依存してエージングスイッチ1 0 2によって制御される際に、別々に書き込まれる。この仕方では、動的テーブルA 0 6 6及び動的テーブルA 1 6 8は、稼働中の10 テーブル及び履歴テーブルの役割を果たすように別々である。の役割を果たすように交互である。タイマが5分後に期限切れになったとき、より古いエントリ（履歴テーブル）を有するテーブルは消去回路1 0 6によって消去される。次いで、履歴テーブルは稼働テーブルとなり、エントリは、次の5分間の間に、その中に書き込まれる。同様に、ネットワークBからのパケットのソースアドレスは、ハッシュ値を得るためにハッシュされ、指示（1ビット）は、ソースアドレスに対応するアドレスがサイドBに存在することを示すために、動的テーブル20 B 0 7 2か動的テーブルB 1 7 4のいずれかにストアされる。また、動的テーブルB 0 7 2及び動的テーブルB 1 7 4は、古い指示をエージングし且つ消去することを行うための指示を書き込むために別々に使用される。静的テーブルA 7 0及び静的テーブルB 7 6は、普段はパケットをブロードキャストしないノードに対応する指示をストアし、従って、通常は、対応する指示を動的テーブルに有していない。CPUは、CPUインターフェース1 6 6を介して指示を静的テーブルA 7 0及び静的テーブルB 7 6に書き込む。判断回路1 2 2は、30 パケットがセグメントA 9 0からセグメントB 1 6 4に渡されるかどうか制御する。パケットがセグメントAからセグメントBに渡されたかどうかは、静的テーブル及び動的テーブルにストアされた指示に基づいて判断回路1 2 2によって判断される。同様に、パケットがセグメントBからセグメントAに渡されたかどうかは、静的テーブル及び動的テーブルの指示に基づいて判断回路1 5 6によって判断される。

【0 0 2 0】パケットがネットワークセグメントから到着したとき、指示は、そのパケットのソースを示すために、対応する動的テーブルにストアされる。パケットの宛先アドレスは、パケットが宛先アドレスに対応する40 テーブルの指示に基づいた別のネットワークセグメントに転送されるべきかどうか判断するのに用いられる。図5は、判断回路及び、本発明によるネットワークAからのパケットのためのパススルー回路の回路図である。ORゲート1 8 4の入力は、アドレスレコーダ1 1 4を介して静的テーブルA 7 0と、アドレスレコーダ1 1 2を介して動的テーブルA 1 6 8と、アドレスレコーダ1 1 0を介して動的テーブルA 0 6 6とに接続されている。ORゲート1 8 0の入力は、アドレスレコーダ1 2

13

0を介して静的テーブルB76と、アドレスレコーダ116を介して動的テーブルB072と、アドレスレコーダ118を介して動的テーブルB174とに接続されている。ORゲート184の出力は、NANDゲート186の入力に接続されている。ORゲート180の出力は、インバータ182の入力に接続されている。インバータ182の出力は、NANDゲート186の入力に接続されている。NANDゲート186の出力は、バススルー回路が、パケットに対応する指示が静的テーブルB76、動的テーブルB072又は動的テーブルB174に設定されているならばパケットを通すことができ、対応する指示が静的テーブルA70、動的テーブルA168及び動的テーブルA066に設定されていなければパケットを通すようにバススルー回路124の制御に接続されている。

【0021】図6は、判断回路及び、本発明によるネットワークBからのパケットのためのバススルー回路の回路図である。ORゲート190の入力は、アドレスレコーダ154を介して静的テーブルA70と、アドレスレコーダ152を介して動的テーブルA168と、アドレスレコーダ150を介して動的テーブルA066とに接続されている。ORゲート194の入力は、アドレスレコーダ148を介して静的テーブルB76と、アドレスレコーダ142を介して動的テーブルB074と、アドレスレコーダ146を介して動的テーブルB174とに接続されている。ORゲート190の出力は、インバータ192の入力に接続されている。ORゲート194の出力は、NANDゲート196の入力に接続されている。NANDゲート196の出力は、バススルー回路158の制御に接続されている。判断回路156は、パケットがネットワークセグメントB164をネットワークセグメントA90に通すかどうかを制御する。

【0022】図7は、本発明によるバススルー規則を図示する。番号0は、指示が特定のテーブルに設定されていないことを表わしており、例えば、A0の下第1のエントリは0である。このことは、対応する指示が現在の設定でないことを表わしている。番号1は、対応する指示が設定されていることを示す。例えば、A0の下第2のエントリは1であり、A0の対応するエントリが設定されていることを示す。文字「X」は、「X」で示される対応する値が関連性のない特定の行に関して示しており、それが0であろうと1であろうと、その結果は同じである。第1行では、(イネーブルカラムENの1によって表わされているように)チャートは割込み可能なバススルーを示し、A0の対応するエントリが設定されていないとき、A1は設定されておらず、静的テーブルAは設定されていない。チャートの第1行に表わされている特定の状態に関して、B0、B1及び静的テーブルの値は、対応するエントリのXによって表わされているように無関係である。従って、セグメントAノードに

(8)

特開平11-4247

14

対応するテーブルが対応する指示で設定されていなければ、パケットはセグメントAからセグメントBに通され、セグメントBに対応するいかなるテーブルもが対応する指示で設定されているならば、パケットはセグメントAからセグメントBに通される。図7の下半分は、パケットをセグメントB164からセグメントA190に通すための規則を示す。図8は、本発明によるテーブルを有するエンティストアと選択回路のブロック図である。アドレスデコーダ100及びアドレスデコーダ110は、マルチプレクサとして各々実行される。ハッシュ回路98は、第1のハッシュ値を生成するためにソースアドレス96をハッシュする。第1のハッシュ値は、エントリを動的テーブルA066に書き込むために、アドレスデコーダ100によって使用される。動的テーブル66に書き込まれたエントリは、ハッシュ回路98によって提供される第1のハッシュ値によって指標付けされる。宛先アドレス94は、第2のハッシュ値を生成するために、ハッシュ回路108によってハッシュされる。第2のハッシュ値は、動的テーブルA066からの宛先アドレス94に対応する指示を選択する際、アドレスデコーダ110によって使用される。6バイトアドレス(MACアドレス)が8ビットハッシュ値内にハッシュするのに使用されるので、2又はそれ以上のアドレスが同じハッシュ値にハッシュする可能性がある。従って、第2ハッシュ値が設定されているならば、それは宛先アドレス94と等しいソースアドレスを有するパケットに対応して設定されることがあり、宛先アドレス94と等しくないが同じ8ビットハッシュ値にハッシュするソースアドレスを有するパケットによって設定され得る。

【0023】図9は、本発明によるテーブルのブロック図である。動的テーブルA066は、ブリッジデバイスによって使用される他のテーブル(動的テーブルA168、動的テーブルB072、動的テーブルB174、静的テーブルA70及び静的テーブルB76)と同じ方法で構築される。テーブルA066は合計256のエントリを含み、各エントリは、アドレスのセットのうちの少なくとも1つのアドレスがブリッジの特定のサイドに存在するかどうかの指示に対応する。256の指示の間の各指示は、1ビット値としてストアされる。各1ビット値は、例えば、フリップフロップ200-0のような、単一のDフリップフロップによってストアされる。従って、動的テーブルA0は、フリップフロップ200-7を介するフリップフロップ200-0と、フリップフロップ202-7を介するフリップフロップ202-0と、合計256のフリップフロップを有する他のフリップフロップを含む。指示は、アドレスデコーダ100によってテーブルA066に選択されるにつれて、ハッシュ回路98によって生成されたハッシュ値によって指標付けされるテーブルA066の位置にフ



リップフロップを設定することによってテーブルA 0 6 6内に書き込まれる。テーブルA 0は、テーブルA 0を含む全てのフリップフロップをクリアすることによって消去される。テーブルは、3 2の8ビットレジスタからなる。図9に示した各カラムは8ビットレジスタからなり、例えば、行はフリップフロップ2 0 0-7を介するフリップフロップ2 0 0-0からなる。リセット後、対応するサイドに受け入れられた全てのバケットが転送される方法で、全てのテーブルは「0」にクリアされる。静的テーブルエントリは設定され、外部CPUによってクリアされる。動的テーブルは選定され、学習及びエージングプロセス中にクリアされる。

【0 0 2 4】図1 0 aは、本発明の実施形態によるハッシュ関数のブロック図である。非マルチキャスト及び非ブロードキャストパケットの宛先アドレス2 1 0の最初の6バイトは、IEEE規格8 0 2. 3イーサネット周期冗長検査(CRC)ファンクション(ブロック2 1 2)を介して送られる。3 2ビットCRC値2 1 4は、CRCファンクション2 1 2から生じる。3 2ビットCRC値2 1 4の最上位の8ビットは、ラッチされる(ブロック2 1 6)。これらの8ビットは、指示(1ビット)をテーブル(例えば、動的テーブルA 0 6 6、動的テーブルA 1 6 8、動的テーブルB 0 7 2又は動的テーブルB 1 7 4)に指標付けするためにデコードされる。デコードのために、最下位の5ビット(アイテム2 1 9)は、テーブル(例えば、動的テーブルA 0 2 1 6)の3 2のレジスタの間の1つのレジスタに指標付けし、最上位の3ビット(アイテム2 2 0)は、選択されたレジスタのビットに指標付けする。図1 0 aに示された例では、宛先アドレスはレジスタ2 1、ビット2

にハッシュするのが分かる。ハッシュ関数はまた、テーブルにストアされた指示をストアするためにソースアドレスをハッシュし、指示を静的テーブル(政敵テーブルA 7 0及び静的テーブルB 7 6)から得るために宛先アドレスをハッシュするために使用される。

【0 0 2 5】図1 0 bは、周期冗長検査(CRC)回路のブロック図である。アドレスビットは、媒体独立インターフェース(MII)4 6 0から得られ、8ビットハッシュ値4 8 2を生ずるためにCRC回路によって処理される。CRC回路は、レジスタ4 6 2及び4 6 4と、排他的論理和(XOR)論理ブロック4 6 6、4 7 6、4 7 8及び4 8 0と、Dフリップフロップ4 6 8、4 7 0、4 7 2及び4 7 4とを含む。MII 4 6 0からのアドレスビットはレジスタ4 6 2によって受け入れられる。レジスタ4 6 2の出力は4ビット値であり、レジスタ4 6 4の入力とXOR 4 6 6の入力とに接続される。レジスタ4 6 4の出力は4ビット値であり、XOR 4 6 6への8ビット入力として、レジスタ4 6 2からの出力と結合される。XOR 4 6 6の出力は、XOR 4 7 6の入力と、XOR 4 7 8の入力と、XOR 4 8 0の入力

と、Dフリップフロップ4 7 4の入力とに接続される。XOR 4 7 6の出力は、Dフリップフロップ4 6 8の入力に接続される。Dフリップフロップ4 6 8の出力は、8ビットハッシュ値4 8 2として、XOR 4 6 6の入力に接続される。XOR 4 7 8の出力は、Dフリップフロップ4 7 0の入力に接続される。Dフリップフロップ4 7 0の出力は、XOR 4 7 6の入力に接続される。XOR 4 8 0の出力はDフリップフロップ4 7 2の入力に接続される。Dフリップフロップ4 7 2の出力はXOR 4 7 8の入力に接続される。Dフリップフロップ4 7 4の出力はXOR 4 8 0の入力に接続される。

【0 0 2 6】CRCハッシュ関数をここに記載したけれども、他の決定性ランダム化関数が、指示をテーブルに指標付けすることによってインデックス値を生成するのに使用されることができるとは明らかである。周期冗長検査の議論に関しては、ローカルエリアネットワークに関するAmerican Standard IEEE規格: ANSI/IEEE Std 802. 3-1985 ISO Draft International Standard 8802/3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, Technical Committee on Computer Communications of the IEEE Computer Society (1 9 8 3年6月2 4日承認、American National Standards Institute 1 9 8 4年1 1月2 1日IEEE規格会議)を参照し、これを参考文献としてここに組み入れる。特にその文献の3. 2. 8章を参照のこと。

【0 0 2 7】図1 1は、本発明によるフィルタリング関数を図示するチャートである。自己学習モードが選択されている1乃至1 6の場合では、ブロックされているパケットの番号が、各個別のケースによってフィルタリングされるそれらと、並びに、自己学習テーブルによって拒否されたそれらとを含む。逆フィルタリングオプションが起動されるならば(ケース1 7, 1 8, 1 9)、ブロードキャスト又はマルチキャストパケットだけが転送され、パケットの残りはフィルタリングオプションの設定にかかわらずブロックされる。ブロードキャストフィルタでは、4 8ビット宛先アドレスに全て「1」を持ったパケットは転送されない。マルチキャストパケットフィルタでは、グループアドレスビットに「1」を持ったパケットは転送されず、これはブロードキャストパケットを含まない。自己アドレッシングパケット(DA=S A)フィルタでは、同じ宛先アドレスとソースアドレスをもったパケットは転送されない。逆ブロードキャストフィルタでは、4 8ビット判断アドレスに全て「1」を持ったパケットだけが転送される。逆マルチキャストパケットフィルタでは、グループアドレスであるがブロードキャストアドレスでないものに「1」を持ったパケットだけが、転送される。

【0 0 2 8】2 5 6ビット静的ハッシュフィルタでは、2つの静的ハッシュフィルタテーブルSTATA及びS

T A T Bがある。ポートAからの非マルチキャスト／ブロードキャストパケットの宛先アドレスは、対応するビットをS T A T A及びS T A T Bに提供するために、上述したようにハッシュされる。S T A T Aに指標付けされたビットが「1」に設定され、且つ、S T A T Bの対応するビットが「1」に設定されないならば、このパケットはポートB内に転送されないようにブロックされる。S T A T Aの指標付けされたビット及びS T A T Bの対応するビットが共に「1」に設定されるならば、パケットはブロックされない。ポートBからポートAにフィルタリングするハッシュは、同じ方法で働く。図12は、本発明によるネットワークAからのパケットに関する自己学習及びフィルタリングを図示するフローチャートである。第1のパケットは、セグメントAから受信される（ステップ246）。テーブルAでは、パケットのソースアドレスに対応する指示がストアされている。次のテーブルAは、テーブルAの指示がパケットの宛先アドレスに対応して存在するかどうか判断するためにチェックされる。パケットへの宛先アドレスに対応するテーブルAの指示が存在しないならば、次いで、パケットはネットワークセグメントBに通される（ステップ256）。パケットの宛先アドレスに対するテーブルAの指示が存在しているならば、次いで、パケットの宛先アドレスに対応するテーブルBの指示が存在するかどうか判断される（ステップ252）。パケットの宛先アドレスに対応するテーブルBの指示が存在しているならば、次いで、パケットはネットワークセグメントBに通される（ステップ256）。パケットの宛先アドレスに対応するテーブルBの指示が存在していないならば、次いで、パケットはブロックされる（ステップ254）。従って、パケットの宛先アドレスに対応するテーブルAの指示が存在していないならば、次いで、テーブルBはチェックされない。さもなければ、テーブルBはチェックされる。同じハッシュ値にハッシュするアドレスのセットの別のアドレスが、セットされ得るテーブルAの指示の原因となっている状況を説明するのを助けるために、パケットの宛先アドレスに対応するテーブルAの指示が存在するならば、テーブルBはチェックされる。

【0029】図13は、本発明によるネットワークBからのパケットに関して自己学習し、フィルタリングすることを図示するフローチャートである。パケットをネットワークセグメントBから受け取る（ステップ258）。指示をパケットのソースアドレスに対応するテーブルAにストアする（ステップ260）。テーブルBは、パケットの宛先アドレスに対応する指示がテーブルBに存在するかどうか判断するためにチェックされる（ステップ262）。パケットの宛先アドレスに対応するテーブルBの指示が存在しないならば、次いで、パケットをネットワークセグメントAに通す（ステップ268）。パケットの宛先アドレスに対応するテーブルBの

指示が存在するならば、次いで、テーブルAは、パケットの宛先アドレスに対応する指示がテーブルAに存在するかどうか判断するためにチェックされる（ステップ264）。パケットの宛先アドレスに対応するテーブルAの指示が存在するならば、次いで、パケットをネットワークセグメントAに通す（ステップ268）。パケットの宛先アドレスに対応するテーブルAの指示が存在しないならば、次いで、パケットをブロックする（ステップ266）。

10 【0030】図14は、本発明によるネットワークセグメントAからのパケットに関する自己学習及びフィルタリングを図示するより詳細なフローチャートである。パケットをサイドAから受け取る（ステップ270）。ソースハッシュ値を形成するために、サイドAからのパケットのソースアドレスをハッシュする（ステップ272）。ソースハッシュ値に基づいてテーブルA0又はテーブルA1にビットを設定する（ステップ274）。次に、宛先ハッシュ値を形成するために、宛先アドレスをハッシュする（ステップ276）。次に、これらのテーブルのいずれもが宛先ハッシュ値に対応する指示ビット

20 セットを有するかどうか判断するために、テーブルA0、A1及びS T A T Aをチェックする。これらのテーブルのいずれもが宛先ハッシュ値に対応するビットセットを有しないならば、次いで、パケットをセグメントBに通す（ステップ284）。テーブルA0、A1又はS T A T Aのうちの少なくとも1つが、宛先ハッシュ値によって指標付けされたエントリにビットセットを有するならば、次いで、宛先ハッシュ値に対応する位置でテーブルB0、B1及びS T A T Bをチェックする（ステップ280）。宛先ハッシュ値によって指標付けされた対応するビットのうちの少なくとも1つがテーブルB0、B1又はS T A T Bに設定されているならば、次いで、パケットをセグメントBに通す（ステップ284）。B0、B1又はS T A T Bのいずれかに宛先ハッシュ値によって指標付けされたビットセットがないならば、次いで、パケットをブロックする（ステップ282）。

30 【0031】ステップ274では、ビットは、エージングタイマに依存するテーブルA0又はテーブルA1のいずれかに設定される。この仕方では、テーブルA0及びA1は、より古い指示のセット及び指示のより新しいセットをストアさせることができる。所定時間（5分）後、より古いテーブルのエントリが消去され、次いで、新しいテーブルが古いテーブルとなり、次いで、新しいエントリが、以前のより古いテーブルに書き込まれ得る。エージングタイマは、フラッシュされる前に、動的ハッシュテーブルを保持するためのインターバル時間を決定するのに使用される。エージングタイマのインターバルは、ピンを調節することによって、又は、本発明の集積回路の実施形態のエージングタイマレジスタをプログラミングすることによって選択される。本発明の一の

50

実施形態では、最小エージング時間は5分であり、最大は1275分である。変形実施形態では、エージングタイムは、古いエントリのフラッシングが生じないように、ターンオフされ得る。他の可能なエージング時間設定を有する本発明の他の実施形態が可能である。

【0032】図15は、本発明によるネットワークBからのバケットに関する自己学習及びフィルタリングを図示するより詳細なフローチャートである。バケットをサイドBから受け取る（ステップ288）。ソースハッシュ値を形成するために、サイドBからのバケットのソースアドレスをハッシュする（ステップ290）。テーブルB0又はテーブルB1のビットが、ステップ290で得られたソースハッシュ値に対応して設定される（ステップ292）。宛先ハッシュ値を形成するために、バケットの宛先アドレスをハッシュする（ステップ294）。宛先ハッシュ値に対応するビットが、テーブルB0、テーブルB1又はテーブルSTATABのいずれかに存在するかどうか判断する。宛先ハッシュ値によって指標付けされた対応する宛先ビットが、テーブルB0、テーブルB1又はテーブルSTATABに設定されていなければ、次いで、バケットをセグメントAに通す（ステップ302）。宛先ハッシュ値によって指標付けされた対応するエントリがテーブルB0、テーブルB1又はSTATABに存在するならば、次いで、それらが宛先ハッシュ値によって指標付けされた指示ビットを含むかどうか判断するために、テーブルA0、テーブルA1及びテーブルSTATAをチェックする（ステップ298）。指示が、テーブルA0、テーブルA1又はSTATAの宛先ハッシュ値によって指標付けされたセットであるならば、次いで、バケットをセグメントAに通す（ステップ302）。宛先ハッシュ値によって指標付けされた指示がテーブルA0、テーブルA1又はSTATAのセットでないならば、次いで、バケットをブロックする（ステップ300）。

【0033】図16は、本発明による媒体アクセス制御（MAC）ブロックを備えるブリッジデバイスの概略ブロック図である。ブリッジデバイス324は、受信MAC326と、転送MAC328と、受信MAC330と、転送MAC332と、ハッシュフィルタ338と、バッファ管理ブロック340と、バッファB334と、バッファA336を含む。受信MAC326は、バケットを第1のネットワークから受け取り、バケットをバッファB334に入れる。転送MAC332は、バケットをバッファB334から第2のネットワークに転送する。受信MAC330は、バケットを第2のネットワークから受け取り、それらのバケットをバッファA336内に入れる。転送MAC328は、バケットをバッファA336から第1のネットワークに転送する。ハッシュフィルタ338は、バケットが、第1のネットワークから第2のネットワークに通されたか、若しくは、第2の

ネットワークから第1のネットワークに通されたかどうかを制御し、かかる第1のネットワークは、バケットの宛先アドレスのハッシュ値に基づき、及び、バケットのアドレスに対応する指示をストアするテーブルに基づく。バッファ管理340は、バッファB334及びバッファA336を管理する。MACは、半二重及び全二重の実施形態で完全にIEEE802.3準拠である。

【0034】図17は、本発明によるバッファの概略図である。バッファ344は、ポートA及びポートBに結合されたネットワークセグメントからのバケットをストアする。メモリは、2つのセクションに分割される。セクションAは、ポートAに関するバッファを受け入れ、セクションBはポートBに関する。バッファ344は64KBのサイズからなる。ポートAからのバケット（例えば、バケット346、348、350）はバッファ344の第1の32KBにストアされる。ネットワークセグメントBからのバケット（例えば、バケット352、354）は、バッファ344の第2の32KBにストアされる。バッファは高速（20nsec又はそれ以上）SRAMで実行される。最小でも16KBのバッファメモリが必要であるが、256KBバッファが、より高性能に関しては好ましい。各セクションのサイズは、外部ピンを介して構成される。変形実施形態では、バッファのセクションが一旦いっぱいになったときに動的バッファ割り当てをすることができるように、自動サイジングを使用することができる。図18は、割り当てがポートAとポートBとの間で1対1であるバッファ344の内部構造を示す。各バッファの最後のワードは状況ストレージである。バケットを受信した後、ブリッジは、バケット長をストレージに書き込む。不良バケットをエンドで拒絶し、バッファを再生する。リセット後、両サイドは100Mbpsで選択され、A及びBのバッファサイズは等しい。最小サイズのバケットバッファは2KBである。バッファメモリは4つのモードを有する：

- (1) Aから受信し、Bに転送する
- (2) Bから受信し、Aに転送する
- (3) Aから受信し、Bから受信する
- (4) Aに転送し、Bに転送する

全てのバケットが受信され、全てのバッファ空間が利用可能でないならば、`buff_full`ピンは、空のバッファが無い限りアサートされる。全二重モードでは、フロー制御機構が使用可能ならば、全てのバッファ空間が利用可能でないことを信号で伝えるために、休止バケットが他のエンドに送信される。他のエンドがバケットを使うことを止めさせるための休止タイムの時間間隔は、シリアルEEPROMによって決定される。

【0035】2つの可能な転送モード及び2つの送信プロトコルが、ブリッジデバイスのピンを介して選択される。転送モードは、完全なバケットストア及び転送、又は、64バイトストア及び転送のいずれかである。完

全なパケットストア及び転送モードでは、完全なパケットが受け取られてチェックされ、不良形成パケットが放棄され、バッファが再生された後に、パケットが転送されるだけである。64バイトストア及び転送モードでは、ブリッジは、一旦最初の64バイトが衝突なしで受信されるならば、受信パケットを転送するが、受信中ならば、別のパケットが伝送され、或いは、伝送されるように待機し、このオプションは、影響を有しない、即ち、受信されたパケットは全部バッファに入れられ得る。半二重伝送プロトコルでは、衝突検出機能を備えたキャリア検知多重アクセス (CSMA/CD) プロトコルが実行される。全二重及び完全制御プロトコルでは、衝突及びキャリア検知検出なしで、伝送は、第64のバイトの位置又はパケットの最後ではじまる。休止パケットが受信され、伝送の使用が、休止時間の満了まで中断されている。

【0036】ポートAに4つのタイプの物理的インターフェースと、ポートBに6つのタイプの物理的インターフェースとが、異なるアプリケーションを収容するために提供される。これらのインターフェースは、10/100 MII データインターフェース、100 M TX/FX インターフェース、及び、10 M シリアルインターフェースである。ポートA及びBの物理的インターフェースは、それぞれのポートに対応するピンを介して独立して選択され得る。フレーム4ビット・ニブル・ワイド同期式データバス及び制御信号を有し、最も一般的に使用されている10/100 MII データインターフェースが、2つのグループに分類されている。MAC-MII

ポートAの物理的インターフェース	
MAC-MII (PHYに接続)	
PHY-MII (MACに接続)	
FX	
TX	
ポートBの物理的インターフェース	
MAC-MII (PHYに接続)	
PHY-MII (MACに接続)	
FX	
TX	
MACシリアル (PHYに接続)	
PHYシリアル (MACに接続)	

100 Mb/s MII インターフェース、10 Mb/s ニブル・インターフェース及び10 Mb/s シリアルインターフェースという3つのタイプの MII インターフェースが提供されている。10/100 Mb/s MII は、2.5/25 MHz クロック速度で4ビット・ニブル・モードで作動する。種々のアプリケーションに関してより柔軟性を提供するために、各 MII アプリケーションに関する受信された MII モードはまた実行される。例えば、MAC-MII インターフェースは PHY デバイスに接続するのに使用されるが、それは受信さ

I インターフェースは、PHY デバイスから RXD [3:0] を介してデータを受け取り、TDX [3:0] を介してデータを送り、PHY-MII インターフェースは MAC のようなデバイス又はリピータデバイスから TDX [3:0] を介してデータを受け取り、RXD [3:0] を介してデータを送る。10 M MII の4ビット・ニブル・モードはまた、データを2.5 MHz クロック速度で伝送することによって使用可能である。100 M TX/FX インターフェースは、情報を MAC 及びリピータへ、及び、MAC 及びリピータから転送することができ、該リピータは25 MHz のクロック速度で下位の待ち時間を有するフレームのない5ビットのデータを使用する。5ビット・コード・グループは検出されず、「ストリーム」と呼ばれる外部100 BASE-X 物理的プロトコルデータユニットを意味しない。ポートBだけが、10 MHz クロック速度でシリアルデータストリームを備えるデータを伝送するために、10 M シリアルモードに構成されうる。10 M シリアルモードの2つのタイプが利用可能であり、MAC シリアルモードによって、「7ワイヤ」インターフェースを10 Base-T PHY にさせることができ、PHY シリアルモードによって、「7ワイヤ」インターフェースを10 Base-T MAC にさせることができる。これらのインターフェースは、MAC シリアルモードに関する (1, 0, 0) 及び PHY シリアルモードに関する (1, 0, 1) で PHY2-0#B を設定することによって選択される。インターフェース選択を以下に詳細に示す：

速度 (Mb/s)	
10/100	
10/100	
100	
100	
速度 (Mb/s)	
10/100	
10/100	
100	
100	
10	
10	

れた MII であり、PHY-MII インターフェースは MAC デバイスに接続するのに使用される。MAC-MII は15の信号を含み、RXDV, CRS, COL, RXER, RXD [3:0] が入力であり、TXEN, TXD [3:0] が出力である。MAC-MII モードでは、物理層チップにソースがあるクロックライン TX CLK 及び RX CLK は、クロック速度が10 MBで2.5 MHz 及び100 MBで25 MHz のブリッジデバイスに関する伝送及び受信クロックを供給する。一方、RXDV, CRS, COL, RXER, RXD

〔3:0〕を出力として、TXEN, TXD〔3:0〕を入力として備えるMACデバイスへのPHY-MIIインターフェースは、PHY-MIIインターフェースを提供する。MX98201及び物理層チップに関するクロックリファレンスを与えるTXCLKは、外部の2.5/25MHzクロックによってソースされ、RXCPHYはブリッジデバイスによってソースされる。10MB/sシリアルインターフェースオペレーションに関して、MIIデータインターフェースは、10MB/sインターフェースを収容するために再配置される。MACシリアルインターフェースは、ブリッジデバイスへの入力としてRXD, CRS, COLと、10MB/sイーサネット物理層への出力としてTXEN, TXDとを提供するのに用いられる。物理層チップによって生成されるTXCLK及びRXCLKクロックラインは、ブリッジデバイスの10MHz伝送及び受信クロックを提供する。一方、PHYシリアルインターフェースは、MACデバイスへの出力としてRXD, CRS, COLと、入力としてTXEN, TXDとを使用する。外部の10MHzクロックは、MX98201及びMACデバイスに関するTXCLKを供給し、RXCPHYはブリッジデバイスによってソースされる。図18aは、本発明による、ブリッジデバイスと、2つの100MB/s衝突ドメインの概略ブロック図である。図18aに示された設計は、両方の衝突ドメインに単一のバンド幅(100MB/s)を使用する。第1の衝突ドメイン358は、DTE364、DTE366、DTE368、及び、DTE370に接続されているリピータ362を含む。第2の衝突ドメイン360は、DTE374、DTE376、DTE378、及び、DTE380に接続されているリピータ372を含む。第1の衝突ドメイン358は、ブリッジデバイス356を介して第2の衝突ドメイン360に接続される。ブリッジデバイスは、第1の衝突ドメイン358のリピータ362と、第2の衝突ドメイン360のリピータ372とに接続される。

【0037】図18bは、本発明による、ブリッジデバイスと、1つの10MB/s衝突ドメイン及び1つの100MB/s衝突ドメインとの概略ブロック図である。図18bに示された設計は、第2の衝突ドメイン360のバンド幅(100MB/s)と比較して、第1の衝突ドメイン382に異なるバンド幅(10MB/s)を使用する。第1の衝突ドメイン382は、ブリッジデバイス356を介して第2の衝突ドメイン360に接続される。第1の衝突ドメインは、ブリッジデバイス356、DTE386、DTE388、DTE390、及び、DTE392に接続されるリピータ384を含む。第2の衝突ドメインは、ブリッジデバイス356、DTE374、DTE376、DTE378、及び、DTE380に接続されるリピータ372を含む。

【0038】図19は、本発明による複合型100Ba

seerTシステムの概略ブロック図である。かかるシステムは、10MB/s及び100MB/sの両サービスを引き渡すためにリピータとブリッジで構成される。ブリッジデバイス356は、SRAM384及びSRAM386に接続される。ブリッジデバイス356は、MACシリアルポート387を介してTXトランシーバ388に接続される。TXトランシーバは10Baseリピータ390に接続される。10BaseリピータはPC292及びPC394に接続される。ブリッジデバイス356は、媒体独立インターフェース(MII)ポート396を介してリピータコントローラ398に接続される。リピータコントローラ398は、TXポート400を介してTXトランシーバ404と、TXポート402を介してTXトランシーバとに接続される。追加のTXポートは、ポート404を介してリピータコントローラ398に接続され得る。TXトランシーバ404は、TX媒体408を介してPC412に接続される。TXトランシーバ406は、TX媒体410を介してPC414に接続される。

【0039】図20aは、本発明による組み込みブロックアプリケーションの概略ブロック図である。ブリッジデバイス356は、SRAM384及びSRAM386に接続される。ブリッジデバイス356は、TXポート416を介してTXクロックリカバリ&トランシーバ418に接続される。TXクロックリカバリ&トランシーバ418はTX媒体420に接続される。ブリッジデバイス356は、MACMII422を介してリピータコントローラ424に接続される。変形待史形態では、PHYMII/MACMIIインターフェースが、MACMII422の代わりに使用され得る。また、PHYMIIインターフェースが、MACMII422の代わりに使用され得る。図20bは、本発明による媒体独立インターフェース(MII)リピータアプリケーションの概略ブロック図である。ブリッジデバイス356は、TXポート426を介してリピータコントローラ428に接続される。リピータコントローラ428は、TXポートを介してTXクロックリカバリ&トランシーバ434に接続される。TXクロックリカバリ&トランシーバ434は、TX媒体436に接続される。ブリッジデバイス356は、TXポート438を介してTXクロックリカバリ&トランシーバ440に接続される。TXクロックリカバリ&トランシーバ440は、TX媒体442に接続される。変形実施形態では、TXポート426の代わりに、PHY-MIIポートを、ブリッジデバイス356をリピータコントローラ428に接続するのに用いることができ、MIIポートを、リピータコントローラ428をTXクロックリカバリ&トランシーバ434に接続するのに用いることができ得る。図20cは、本発明によるスタンドアロン・ブリッジアプリケーションの概略ブロック図である。ブリッジバイ

ス 3 5 6 は、SRAM 3 8 4 及び SRAM 3 8 6 に接続される。ブリッジデバイス 3 5 6 は、TX ポート 4 4 4 を介して TX クロックリカバリ & トランシーバ 4 4 6 に接続される。TX クロックリカバリ & トランシーバ 4 4 6 は、TX 媒体 4 4 8 に接続される。ブリッジデバイス 3 5 6 は、TX ポート 4 5 0 を介して TX クロックリカバリ & トランシーバ 4 5 2 に接続される。TX クロックリカバリ & トランシーバ 4 5 2 は、TX 媒体 4 5 4 に接続される。変形実施形態では、TX クロックリカバリ & トランシーバ 4 5 2 の代わりに、ブリッジデバイス 3 5 6 を TX クロックリカバリチップに接続することができ、次いで、TX クロックリカバリチップが FX ファイ

パートランシーバに接続され得ることがあり、FX ファイパートランシーバが FX 媒体に接続されることがあり得る。また、TX ポート 4 5 0 及び TX クロックリカバリ & トランシーバ 4 5 2 を使用する代わりに、ブリッジデバイス 3 5 6 が MII ポートを介して 10/100 Base ツイストペア・トランシーバに接続されてよく、次いで、10/100 Base ツイストペア・トランシーバは 10 Base-T 媒体に接続され得る。

【0040】以下は、本発明の実施形態における集積回路のピンを示す表である。

【0041】

【表 1】

ポート A TX シンボル/媒体独立インターフェース (その 1)

ピン名	タイプ	記述
TDAT4_A	O	TX/FX モード：出力。5 ビットシンボルデータの最上位ビット
TDAT3-0_A/TXD3-0_A	I/O	TX/FX モード TDAT3-0_A：出力。これら 4 つのピン及び TDAT4_A は TXCLK_A と同期的な 5 ビットシンボルデータである。MAC-MII モード TXD3-0_A：出力。TXCLK_A と同期的であり、外部 PHY 装置へデータを送信するのに 4 ビット列ニブルが使用される。PHY-MII モード TXD3-0_A：入力。TXCLK_A と同期的であり、4 ビット列ニブルは 外部 MAC 類似装置によって送信されたデータである。
TXCLK_A	I	TX/FX モード：入力。25MHz の局所シンボルクロック入力である。MII モード：入力。2.5/25MHz MII 送信クロック入力である。
RDAT4_A	I	TX/FX モード：入力。5 ビットシンボルデータの最上位ビット
RDAT3-0_A/RXD3-0_A	I/O	TX/FX モード RDAT3-0_A：入力。これら 4 つのピン及び RDAT4_A は RXCLK_A と同期的な 5 ビットシンボルデータである。MAC-MII モード RXD3-0_A：入力。RXCLK_A と同期的であり、外部 PHY 装置からデータを受信するのに 4 ビット列ニブルが使用される。PHY-MII モード RXD3-0_A：出力。RXCLK_A と同期的であり、外部 MAC 類似装置へデータを送信するのに 4 ビット列ニブルが使用される。OE_A がハイであると表明した場合、これらのピンはトライステートである。
RXCLK_A	I	TX/FX モード：入力。25MHz 受信クロックは外部 PHY 装置によって生成される。MAC-MII モード：入力。外部 PHY 装置によって生成された受信クロックは、周波数が 2.5/25MHz である。

【0042】

【表 2】

## ポート A TXシンボル/媒体独立インターフェース (その2)

ピン名	タイプ	記述
SIG-DET_A/COL_A	I/O	TX/FX モード：入力。外部 PHY 装置によって表明した時、信号検出は受信チャネル上の 100 ベース T アクティビティを表す。MAC-MII モード：入力。外部 PHY 装置からの衝突信号は、送信中、受信キャリアが検出されたことを表す。PHY-MII モード：出力。衝突信号は外部 MAC 類似装置とのインターフェースをとり、送信及び受信回線双方が半二重モードで活動状態にある場合、表明する。信号は全二重モードにおいては効果がない。
CRS_A	I/O	MAC-MII モード：入力。キャリア検知信号は、送信または受信回線のいずれかが活動状態にあることを表示するのに、外部 PHY 装置によって発信される。PHY-MII モード。出力。キャリア検知信号は外部 MAC 類似装置とのインターフェースをとる。注記：このモードでは、CRS_A 及び RXDV_A は集積回路内において同ドライブを共有する。
OE_A	I	出力可能。活動状態ロー。ポート A に対する出力として使用するための PHY_MII モードにおける RXD[3:0], RXDV, RXER, RXCPHY_A を制御するのに使用される。
RXCPHY_A	0	PHY_MII モード：出力。2.5/25 MHz クロックソースである。OE_A がハイであると表明した場合、このピンはトライステートである。
RXDV_A	I/O	MAC-MII モード。入力。受信データ有効信号は外部 PHY 装置によって生成される。これは、回復し、複合化されたデータニブルが受信データ回線上にあることを示す。PHY-MII モード。出力。受信データ有効信号は、外部 MAC 類似装置へ集積回路が送信のために有効データニブルを送っていることを示す。このピンは、OE_A をハイに引っ張ることによりトライステートになりうる。注記：このモードでは、CRS_A 及び RXDV_A は集積回路内において同ドライブを共有する。

【0043】

【表3】

## ポート A TXシンボル/媒体独立インターフェース (その3)

ピン名	タイプ	記述
RXBR_A	I/O	MAC-MII モード。入力。受信エラー信号は外部 PHY 装置によって生成される。PHY-MII モード。出力。受信エラー信号は外部 MAC 類似装置に接続する。このピンは OE_A をハイに引っ張ることによりトリステートになりうる。注記：このモードでは、集積回路は常に、集積回路から受信エラーが生成されていないことを示すロジックゼロを駆動する。
TXEN_A	I/O	MAC-MII モード。出力。送信可能は、集積回路が外部 PHY 装置へ送信のために有効データニブルを送っていることを示す。PHY-MII モード。入力。送信可能信号は、回復し、複合化されたデータニブルが外部 MAC 類似装置からのデータ回線上で利用できることを表す。

【0044】

【表4】

## ポート B TXシンボル/媒体独立インターフェース (その1)

ピン名	タイプ	記述
TDAT4_B	0	TX/FX モード：出力。5 ビットシンボルデータの最上位ビット。
TDAT3-0_B/TXD3-0_B	I/O	TX/FX モード TDAT3-0_B：出力。これら 4 つのピン及び TDAT4_B は TXCLK_B と同期的な 5 ビットシンボルデータである。MAC-MII モード：TXD3-0_B：出力。TXCLK_B と同期的であり、外部 PHY 装置へデータを送信するのに 4 ビット列ニブルが使用される。PHY-MII モード：TXD3-0_B 入力。TXCLK_B と同期的であり、4 ビット列ニブルは外部 MAC 類似装置によって送信されたデータである。 MAC シリアルモード TXD0_B：出力。外部 PHY 装置とインターフェースをとるシリアル送信データ。PHY シリアルモード TXD0_B：入力。外部 MAC 類似装置とインターフェースをとるシリアルデータビット。
TXCLK_B	I	TX/FX モード：入力。25MHz の局所シンボルクロック入力である。MII モード：入力。2.5/25MHz MII 送信クロック入力である。 MAC/PHY-MII シリアルモード：10 ベースシリアルモードのための 10 MHz クロックである。
RDAT4_B	I	TX/FX モード：入力。5 ビットシンボルデータの最上位ビット

【0045】

【表5】



## ポート B TXシンボル/媒体独立インターフェース (その2)

ピン名	タイプ	記述
RDAT3- 0_B/RXD3-0_B	I/O	TX/FX モード RDAT3-0_B: 入力。これら 4 つのピン及び RDAT4_B は RXCLK_B と同期的な 5 ビットシンボルデータである。MAC-MII モード RXD3-0_B: 入力。RXCLK_B と同期的であり、外部 PHY 装置からデータを受信するのに 4 ビット列ニブルが使用される。PHY-MII モード RXD3-0_B: 出力。RXCLK_B と同期的であり、外部 MAC 類似装置へデータを送信するのに 4 ビット列ニブルが使用される。OE_B がハイであると表明した場合、これらのピンはトライステートである。
		MAC シリアルモード RXD0_B: 入力。受信シリアルデータビットは外部 PHY 装置とインターフェースをとる。PHY シリアルモード RXD0_B: 出力。シリアルデータビットは外部 MAC 装置とインターフェースをとる。
RXCLK_B	I	TX/FX モード: 入力。25MHz 受信クロックは外部 PHY 装置によって生成される。MAC-MII モード: 入力。外部 PHY 装置によって生成された受信クロックは、周波数が 2.5/25MHz である。
		MAC シリアルモード: 入力。外部 PHY 装置によってソースされた 10MHz 受信クロックである。
SIG- DET_B/COL_B	I/O	TX/FX モード: 入力。外部 PHY 装置によって表明した時、信号検出は受信チャネル上の 100 ベース T アクティビティを表す。MAC-MII モード: 入力。外部 PHY 装置からの衝突信号は、送信中、受信キャリアが検出されたことを表す。PHY-MII モード: 出力。衝突信号は外部 MAC 類似装置とのインターフェースをとり、送信及び受信回線双方が半二重モードで活動状態にある場合、表明する。信号は全二重モードにおいては効果がない。
		MAC シリアルモード: 入力。衝突信号が外部 PHY 装置によって提供される。PHY シリアルモード: 出力。衝突信号は外部 MAC 類似装置とインターフェースをとる。

【0046】

【表6】

## ポート B TXシンボル/媒体独立インターフェース (その 3)

ピン名	タイプ	記述
CRS_B	I/O	MAC-MII モード：入力。キャリア検知信号は、送信または受信回線のいずれかが活動状態にあることを表示するのに、外部 PHY 装置によって発信される。PHY-MII モード。出力。キャリア検知信号は外部 MAC 類似装置とのインターフェースをとる。注記：このモードでは、CRS_B 及び RXDV_B は集積回路内において同ードライバを共有する。
		MAC シリアルモード：入力。キャリア検知信号は外部 PHY 装置とインターフェースをとる。PHY シリアルモード：出力。このキャリア検知信号は外部 MAC 類似装置とインターフェースをとる。注記：このモードでは、CRS_B 及び RXDV_B は集積回路内において同一のドライバを共有する。
OE_B	I	出力可能。活動状態ロー。ポート B に対する出力として使用するための PHY_MII モードにおける RXD[3:0], RXDV, RXER, RXCPHY_B を制御するのに使用される。
RXCPHY_B	0	PHY_MII モード：出力。2.5/25 MHz クロックソースである。OE_B がハイであると表明した場合、このピンはトリステートである。
		PHY シリアルモード。出力。MAC 類似装置（即ち MAC-MII リピータ）とインターフェースをとる時、集積回路及び MAC-MII リピータ双方の RXCPHY 及び TXCLK は同一の外部 10MHz クロックによってソースされる。
RXDV_B	I/O	MAC-MII モード。入力。受信データ有効信号は外部 PHY 装置によって生成される。これは、回復し、複合化されたデータニブルが受信データ回線にあることを示す。PHY-MII モード。出力。受信データ有効信号は、外部 MAC 類似装置へ集積回路が送信のために有効データニブルを送っていることを示す。このピンは、OE_B をハイに引っ張ることによりトリステートになりうる。注記：このモードでは、CRS_B 及び RXDV_B は集積回路内において同ードライバを共有する。

【0047】

【表 7】

## ポート B TXシンボル/媒体独立インターフェース (その 4)

ピン名	タイプ	記述
RXER_B	I/O	MAC-MII モード。入力。受信エラー信号は外部 PHY 装置によって生成される。PHY-MII モード。出力。受信エラー信号は外部 MAC 類似装置に接続する。このピンは OE_B をハイに引っ張ることによりトライステートになりうる。注記：このモードでは、集積回路は常に、集積回路から受信エラーが生成されていないことを示すロジックゼロを駆動する。
TXEN_B	I/O	MAC-MII モード。出力。送信可能は、集積回路が外部 PHY 装置へ送信のために有効データニブルを送っていることを示す。PHY-MII モード。入力。送信可能信号は、回復し、複合化されたデータニブルが外部 MAC 類似装置からのデータ回線上で利用できることを表す。 MAC シリアルモード：出力。送信可能信号は外部 PHY 装置とインターフェースをとる。PHY シリアルモード：入力。送信可能信号は外部 MAC 類似装置とインターフェースをとる。

【 0 0 4 8 】

【表 8】

## モード構成ピン (その 1)

ピン名	タイプ	記述
MSIZE1-0	I	メモリサイズ。ABYTE1-0_B とともにシステム多重化。[MSIZE1,MSIZE0]=メモリサイズ[0,0]=16K バイト; [0,1]=64K バイト; [1,0]=128K バイト; [1,1]=256K バイト。
MRAT2-0	I	メモリ比。これら 2 つのピンはポート A 対ポート B のバッファサイズ比を構成する。ポート A の大きい方のバッファは、ポート A がポート B に送られるより多くのパケットを受信することが可能である。[MRAT2,MRAT0]=ポート A バッファサイズ:ポート B バッファサイズ[x,x,0]=1:1; [0,0,1]=1:7; [0,0,1]=1:15; [1,0,1]=7:1; [1,1,1]=15:1。
AUTOSIZE	I	オートサイズ。ABYTE2_A。ハイであると表明した場合、バッファメモリのオートサイズ機能を可能にする。
FWD1-0_A	I	ポート A フォワードモード選択。[FWD1_A,FWD0_A]=フォワードモード[0,0]=半二重、全パケット蓄積; [0,1]=全二重、全パケット蓄積; [1,0]=半二重、64 バイト蓄積; [1,1]=全二重、64 バイト蓄積。
FWD1-0_B	I	ポート B フォワードモード選択。[FWD1_B,FWD0_B]=フォワードモード[0,0]=半二重、全パケット蓄積; [0,1]=全二重、全パケット蓄積; [1,0]=半二重、64 バイト蓄積; [1,1]=全二重、64 バイト蓄積。
SPEED_A	I	ポート A スピード。ハイであると表明した場合、ポート A には 100M スピードが採用される。ローであると表明した場合、10M スピードが採用される。
SPEED_B	I	ポート B スピード。ハイであると表明した場合、ポート B には 100M スピードが採用される。ローであると表明した場合、10M スピードが採用される。

【0049】

【表 9】

## モード構成ピン (その2)

ピン名	タイプ	記述
PHY1-0_A	I	ポート A の PHY。ABYTE1-0_A とともにシステム多重化。[PHY1_A,PHY0_A]=[0,0]=MAC-MII (外部 PHY 装置とインターフェースをとる); [0,1]=PHY-MII (外部 MAC 装置とインターフェースをとる); [1,0]=100BASE-FX; [1,1]=100BASE-TX。
PHY2-0_B	I	ポート B の PHY。 [PHY2_B,PHY1_B,PHY0_B]=[0,0,0]=MAC-MII (外部 PHY 装置とインターフェースをとる); [0,1,0]=100BASE-FX; [0,1,1]=100BASE-TX; [1,0,0]=MAC シリアル (外部 PHY 装置とインターフェースをとる); [1,0,1]=PHY シリアル (外部 MAC 装置とインターフェースをとる); [1,1,0]=予約済み; [1,1,1]=予約済み。
FCEN_A	I	フロー制御可能。Buful_A とともにシステム多重化。ハイであると表明した場合、ポート A 上の全二重モードにおいてフロー制御メカニズムを可能にする。
FCEN_B	I	フロー制御可能。Buful_B とともにシステム多重化。ハイであると表明した場合、ポート B 上の全二重モードにおいてフロー制御メカニズムを可能にする。

【0050】

【表10】

## アドレス・フィルタ・ピン

ピン名	タイプ	記述
SLENF_AB	I	PKDT7_B とともにシステム多重化。
BCF_AB	I	PKDT6_B とともにシステム多重化。
MCF_AB	I	PKDT5_B とともにシステム多重化。
SAP_AB	I	PKDT4_B とともにシステム多重化。
SLENF_BA	I	PKDT3_B とともにシステム多重化。
BCF_BA	I	PKDT2_B とともにシステム多重化。
MCF_BA	I	PKDT1_B とともにシステム多重化。
SAP_BA	I	PKDT0_B とともにシステム多重化。

【0051】

【表11】

## バッファ・SRAM・インターフェース

ピン名	タイプ	記述
SA16-0	0	SRAM アドレス 16-0。これら 17 のアドレスは SRAM 内における 128K16 ビットの語の 1 つを選択する。
SD15-0	I/O	SRAM データ入力/出力ポート。これら 16 の双方向ピンは SRAM からデータを読み込む、又は SRAM へ書き込むのに使用される。
SR_W	0	書き込まず読み込む。ハイであると表明した場合、SRAMs の読み込み操作を可能にする。ローであると表明した場合、書き込み操作を可能にする。
SG_	0	出力可能。SRAMs からの出力を制御するアクティブな低信号である。

【0052】

【表 12】

## レジスタ・インターフェース

ピン名	タイプ	記述
REGDAT[7:0]	I/O	レジスタデータ。これらは 8 ビットレジスタデータである。CS_ が表明しない場合、トライステートである。
REGSEL[7:0]	I	レジスタ選択。これらピンは内部レジスタを選択するのに使用される。
CS_	I	チップ選択。ローであると表明した場合、レジスタへの読み込み又は書き込みアクセスを可能にする。
REGR_W	I	書き込まず読み込むのをレジスタする。ハイであると表明した場合、読み込み操作を可能にする。ローであると表明した場合、書き込み操作を可能にする。

【0053】

【表 13】

## EEPROM/インターフェース・ピン

ピン名	タイプ	記述
EECS/NOEPRO M	I/O	EEPROM チップ選択。出力。EEPROM 読み込みサイクルの間、ハイであると表明される。 NOEPRO M。入力。EEPROM がない時、ハイに引っ張るべきである。
EESK	0	EEPROM クロック。
EEDI	0	EEPROM シリアルデータ入力。EEPROM の中のシリアルデータに接続される。
EEDO	I	EEPROM シリアルデータ出力。EEPROM の外のシリアルデータに接続される。

【0054】

【表 14】

アドレス・ファイル・アクセス・ピン (その 1)

ピン名	タイプ	記述
PKTD7-0_A/AGTIM7-0	I/O	ポート A からのパケットデータ。PKTD7 はバイトで表すと MSB である。表示される配列は次のとおりである：アイドル・プリアンプル SFD (1 バイト) -DA (6 バイト) -ハッシュされた DA (1 バイト) -SA (6 バイト) ハッシュされた SA (1 バイト) -データ-アイドル。 AGTIM7-0。入力。
ABYTE2_A/AUTO-SIZE ABYTE1_A/PHY1_A ABYTE0_A/PHY0_A	I/O, TTL	ポート A からのアドレスバイト。出力。これら 3 つのピンは PKTD7-0_A の内容を示す。 [ABYTE2, ABYTE1, ABYTE0][0, 0, 0]=PKTD7-0_A は SFD (10101011) である； [0, 0, 1]=PKTD7-0_A は 6DA バイトの CRC の最上位の 8 ビットである； [0, 1, 1]=PKTD7-0_A は SA バイトである； [1, 0, 0]=PKTD7-0_A は 6SA バイトの CRC の最上位の 8 ビットである； [1, 0, 1]=PKTD7-0_A はデータである； [1, 1, 0]=PKTD7-0_A はアイドル状態である； [1, 1, 1]=PKTD7-0_A はプリアンプルである。 オートサイズ。入力。ハイであると表明した場合、バッファメモリのオートサイズ機能を可能にする。PHY1-0_A。ポート A への物理的インターフェース。
ABORT_A	I	ポート A からのパケット放棄。このピンがハイであると表明した場合、最新受信パケットを放棄するように集積回路に信号を送る。

[ 0 0 5 5 ]

[表 1 5]

## アドレス・ファイル・アクセス・ピン (その 2)

ピン名	タイプ	記述
PKTD7-0_B	I/O	ポート B からのパケットデータ。PKTD7 はバイトで表すと MSB である。表示される配列は次のとおりである：アイドル・プリアンプル SFD (1 バイト) -DA (6 バイト) -ハッシュされた DA (1 バイト) -SA (6 バイト) ハッシュされた SA (1 バイト) -データ-アイドル。
SLENF_AB BCF_AB MCF_AB SAF_AB  SLENF_BA BCF_BA MCF_BA SAF_BA		入力。活動状態ハイ信号。A から B への自己学習アルゴリズム可能。 A から B へのブロードキャストフィルタ可能。 A から B へのマルチキャストフィルタ可能。 A から B への自己アドレス指定パケットフィルタ可能。 B から A への自己学習アルゴリズム可能。 B から A へのブロードキャストフィルタ可能。 B から A へのマルチキャストフィルタ可能。 B から A への自己アドレス指定パケットフィルタ可能。
ABYTE2_B ABYTE1_B/MSI ZE1 ABYTE0_B/MSI ZE0	0 I/O	ポート A からのアドレスバイト。出力。これら 3 つのピンは PKTD7-0_A の内容を示す。[ABYTE2, ABYTE1, ABYTE0][0,0,0]=PKTD7-0_A は SFD(10101011)である；[0,0,1]=PKTD7-0_A は 6DA バイトの CRC の最上位の 8 ビットである；[0,1,1]=PKTD7-0_A は SA バイトである；[1,0,0]=PKTD7-0_A は 6SA バイトの CRC の最上位の 8 ビットである；[1,0,1]=PKTD7-0_A はデータである；[1,1,0]=PKTD7-0_A はアイドル状態である；[1,1,1]=PKTD7-0_A はプリアンプルである。MSIZE1-0。入力。これら 2 つのピンはバッファメモリサイズを選択する。
ABORT_B	I	ポート B からのパケット放棄。このピンがハイであると表明した場合、最新受信パケットを放棄するように集積回路に信号を送る。

【 0 0 5 6 】

【表 1 6】



バッファ・フル・インジケーション

ピン名	タイプ	記述
BUFUL_A/FCEN_A	I/O	ポート A においてバッファが一杯である。BUFUL_B がハイであると表明した場合、ポート A の受信パケットに拠りバッファが一杯であることを示す。望まれれば、ポート A バッファ状態インジケーションとして LED を駆動するのに 74LS244 とともに使用されても良い。フロー制御可能。入力。フロー制御メカニズム可能ピン。
BUFUL_B/FCEN_B	I/O	ポート B においてバッファが一杯である。BUFUL_B がハイであると表明した場合、ポート B の受信パケットに拠りバッファが一杯であることを示す。望まれれば、ポート B バッファ状態インジケーションとして LED を駆動するのに 74LS244 とともに使用されても良い。フロー制御可能。入力。フロー制御メカニズム可能ピン。

要するに、本発明は、ブリッジに関する複雑性及びコストを低減するのを助け、ネットワーク媒体の間の相互接続の相関関係をより効率的に提供するのを助け、他のネットワークセグメントへのローカルパケットのリークを避けるのを助け、より高速なネットワークにスケラブルであるデバイスを提供する。前述の本発明の好ましい実施形態の記載は、図示及び記述の目的のために示されている。本発明を開示した正確な形態に限定すべきではない。明らかに、多くの修正及び変形が当業者にとって明らかであろう。本発明の範囲は以下の特許請求の範囲及びその均等の範囲で定義されるべきである。

【図面の簡単な説明】

【図 1】本発明のネットワークセグメント及びブリッジデバイスの概略ブロック図である。

【図 2】本発明によるパケットを通すためのブロックを示すブリッジデバイスの概略ブロック図である。

【図 3】本発明による学習のためのブロックを示すブリッジデバイスの概略ブロック図である。

【図 4】本発明によるブリッジデバイスの更に詳細なブロック図である。

【図 5】本発明によるネットワーク A からのパケットに関する判断回路及びパススルー回路の回路図である。

【図 6】本発明によるネットワーク B からのパケットに関する判断回路及びパススルー回路の回路図である。

【図 7】本発明によるパススルー規則を図示する。

【図 8】本発明によるテーブルを備えた選択回路とエントリストアのブロック図である。

【図 9】本発明によるテーブルのブロック図である。

【図 10 a】ハッシュ関数のブロック図である。

【図 10 b】周期冗長検査回路のブロック図である。

【図 11】本発明によるフィルタリング関数を示すチャートである。

【図 12】本発明によるネットワーク A からのパケット

に関する自己学習及びフィルタリングを図示するフローチャートである。

【図 13】本発明によるネットワーク B からのパケットに関する自己学習及びフィルタリングを図示するフローチャートである。

【図 14】本発明によるネットワーク A からのパケットに関する自己学習及びフィルタリングを図示する更に詳細なフローチャートである。

【図 15】本発明によるネットワーク B からのパケットに関する自己学習及びフィルタリングを図示する更に詳細なフローチャートである。

【図 16】本発明による媒体アクセス制御 (MAC) ブロックを備えるブリッジデバイスの概略ブロック図である。

【図 17】本発明によるバッファの概略図である。

【図 18 a】本発明によるブリッジデバイス及び 2 つの 100 MB/s 衝突ドメインの概略ブロック図である。

【図 18 b】本発明によるブリッジデバイスと、1 つの 10 MB/s 衝突ドメインと、1 つの 100 MB/s 衝突ドメインとの概略ブロック図である。

【図 19】本発明による複合型 100 Base-T システムの概略ブロック図である。

【図 20 a】本発明によるビルドイン・ブリッジ・アプリケーションの概略ブロック図である。

【図 20 b】本発明による媒体独立インターフェース (MII) リピータ・アプリケーションの概略ブロック図である。

【図 20 c】本発明によるスタンドアローン・ブリッジ・アプリケーションの概略ブロック図である。

【符号の説明】

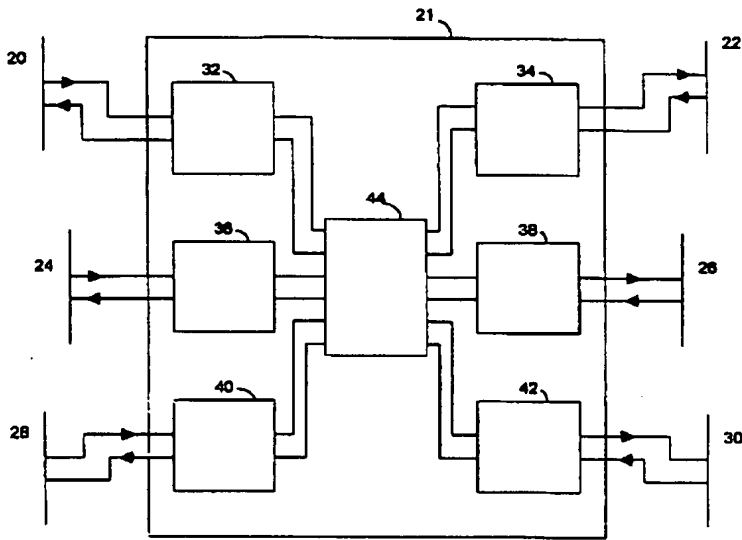
20、22、24、26、28、30 ネットワーク・セグメント

21 ブリッジ・デバイス

- 49
- 3 2、3 4、3 6、3 8、4 0、4 2   ポート
- 4 4   フィルタ
- 4 5   ブリッジ・デバイス
- 4 6、4 8、5 0、5 2   ネットワーク・ノード
- 5 3   ネットワーク・セグメント
- 5 4、5 6、5 8、6 0、6 2   ネットワーク・ノード
- 5 5   ネットワーク・セグメント
- 6 4   フィルタ
- 7 8、8 2   ソース・アドレス
- 8 0、8 4   宛先アドレス
- 8 6、8 8   ハッシュ回路
- 1 0 0   アドレス・デコーダ

- 50
- 1 0 2   エージング・スイッチ
- 1 2 2   判断回路
- 1 2 4   スイッチ
- 1 2 6   バッファ
- 1 3 8   タイマ
- 1 4 0   消去回路
- 1 6 6   CPUインターフェース
- 3 5 6   ブリッジ・デバイス
- 3 5 8、3 6 0   衝突ドメイン
- 10 3 7 2   リピータ
- 4 6 0   M I I
- 4 6 2、4 6 4   レジスタ

【図 1】



【図 7】

X = 0 or 1

(1=B→A)

A→B

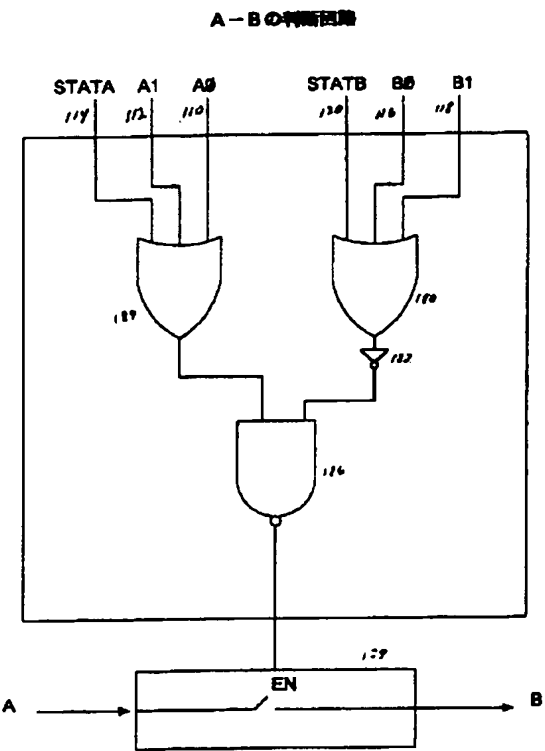
A0	A1	STATA	B0	B1	STATB	EN
0	0	0	X	X	X	1
1	X	X	0	0	0	0
X	1	X	0	0	0	0
X	X	1	0	0	0	0
X	X	X	1	X	X	1
X	X	X	X	1	X	1
X	X	X	X	X	1	1

B→A

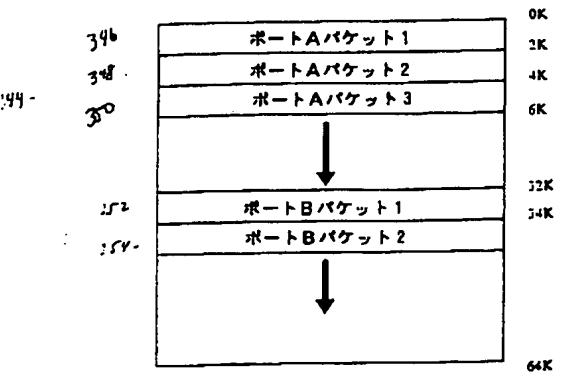
(1=A→B)

B0	B1	STATB	A0	A1	STATA	EN
0	0	0	X	X	X	1
1	X	X	0	0	0	0
X	1	X	0	0	0	0
X	X	1	0	0	0	0
X	X	X	1	X	X	1
X	X	X	X	1	X	1
X	X	X	X	X	1	1

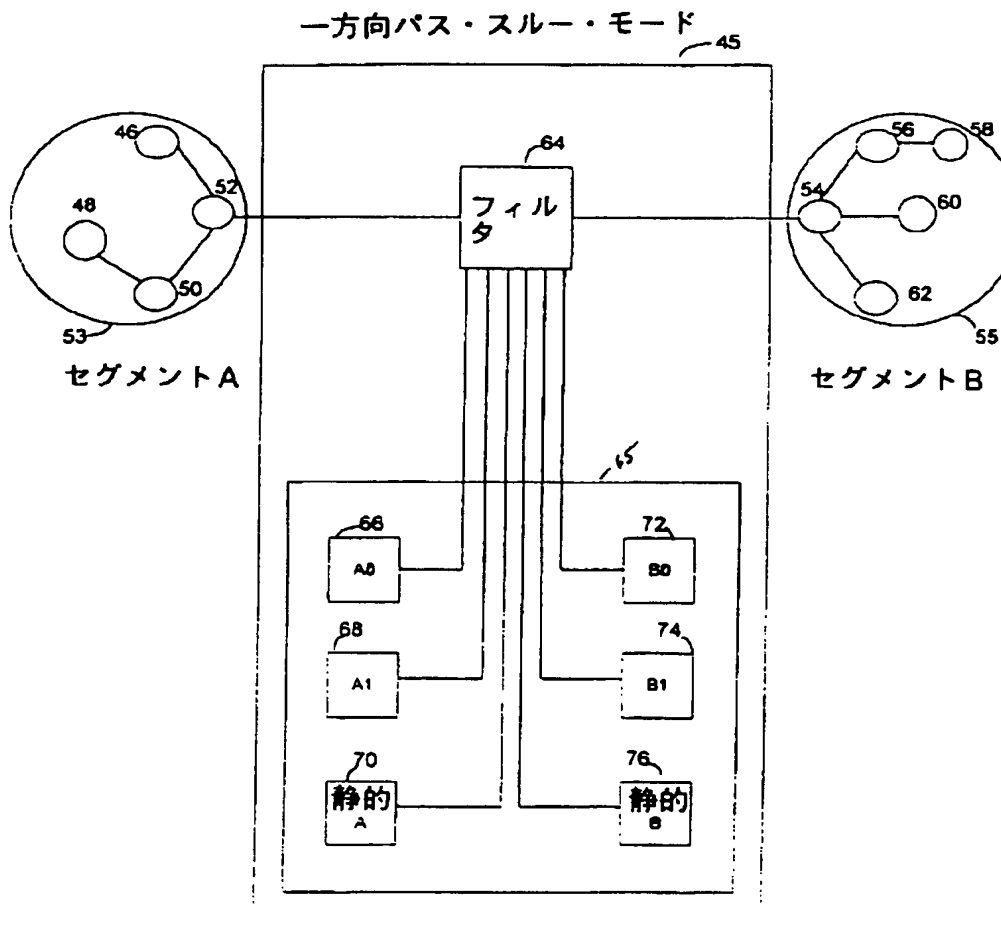
【図 5】



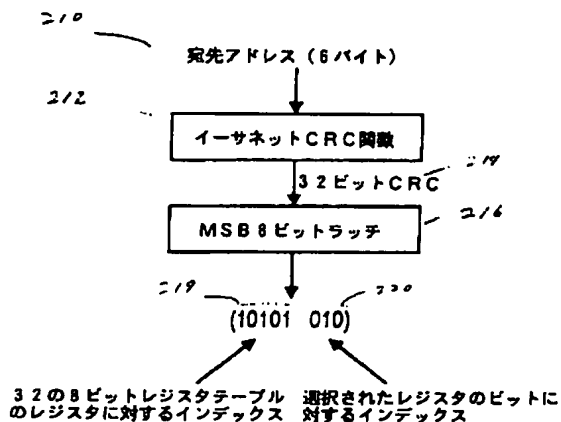
【図 1 7】



【図 2】



【図 10 a】



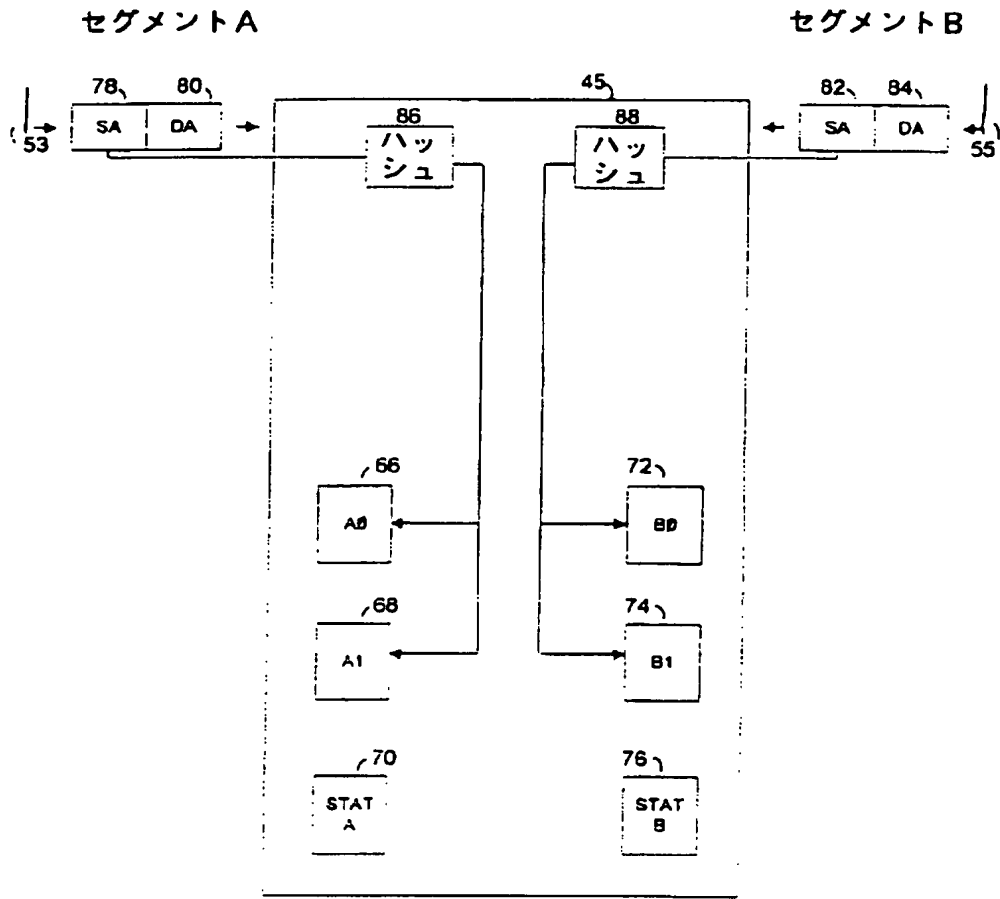
【図 11】

	B	M	S	IB	IM	H	ブロックされるべきパケット
1	0	0	0	0	0	0	なし
2	0	0	0	0	0	1	ブロードキャスト/マルチキャストを除く静的ハッシュ・フィッシング・パケット
3	1	0	0	0	0	0	ブロードキャスト・パケット
4	1	0	0	0	0	1	ブロードキャスト及び静的ハッシュ・フィッシング・パケット
5	0	1	0	0	0	0	マルチキャスト(しかしブロードキャストでない)パケット
6	0	1	0	0	0	1	マルチキャスト(しかしブロードキャストでない)及び静的ハッシュ・フィッシング・パケット
7	0	0	1	0	0	0	自己アドレッシング・パケット
8	0	0	1	0	0	1	自己アドレッシング及び静的ハッシュ・フィッシング・パケット
9	1	1	0	0	0	0	ブロードキャスト及びマルチキャスト・パケット
10	1	1	0	0	0	1	ブロードキャスト、マルチキャスト及び静的ハッシュ・フィッシング・パケット
11	1	0	1	0	0	0	ブロードキャスト及び自己アドレッシング・パケット
12	1	0	1	0	0	1	ブロードキャスト、自己アドレッシング及び静的ハッシュ・フィッシング・パケット
13	0	1	1	0	0	0	マルチキャスト(しかしブロードキャストでない)、自己アドレッシング・パケット
14	0	1	1	0	0	1	マルチキャスト(しかしブロードキャストでない)、自己アドレッシング及び静的ハッシュ・フィッシング・パケット
15	1	1	1	0	0	0	ブロードキャスト、マルチキャスト及び自己アドレッシング・パケット
16	1	1	1	0	0	1	ブロードキャスト、マルチキャスト、自己アドレッシング及び自己ハッシュ・フィッシング・パケット
17	x	x	x	1	0	x	DAI=FFFFFFFFFFFFを有するパケット
18	x	x	x	0	1	x	ブロードキャスト、及び DAI=10 第 10 ビットに 0 を有するパケット
19	x	x	x	1	1	x	DAI=1 の第 1 のビットを有するパケット

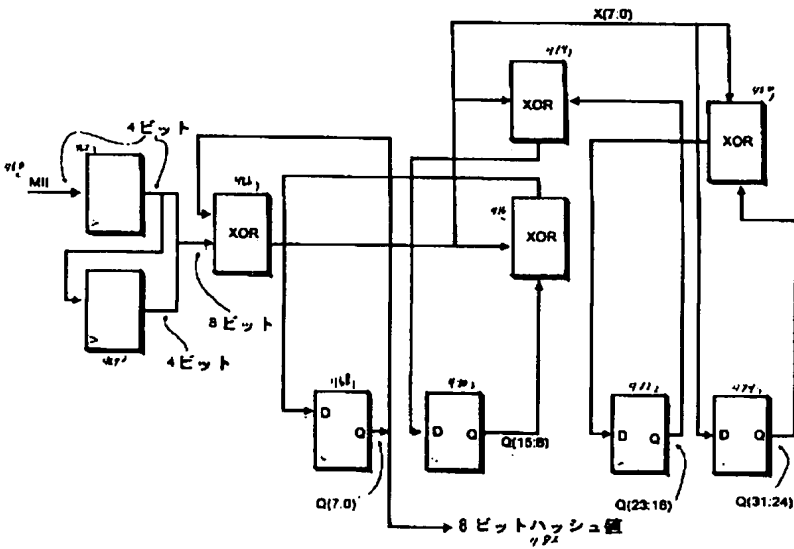
注記: B:ブロードキャスト・パケット・フィッシュ  
 M:マルチキャスト・パケット・フィッシュ  
 S:自己アドレッシング・パケット・フィッシュ  
 IB:逆ブロードキャスト・フィッシュ  
 IM:逆マルチキャスト・パケット  
 H:256ビット・ハッシュ・フィッシュ

【図 3】

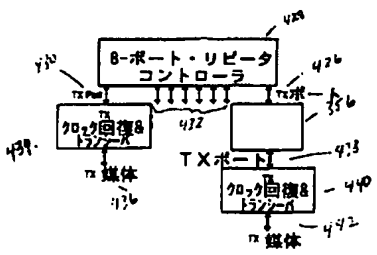
学習モード



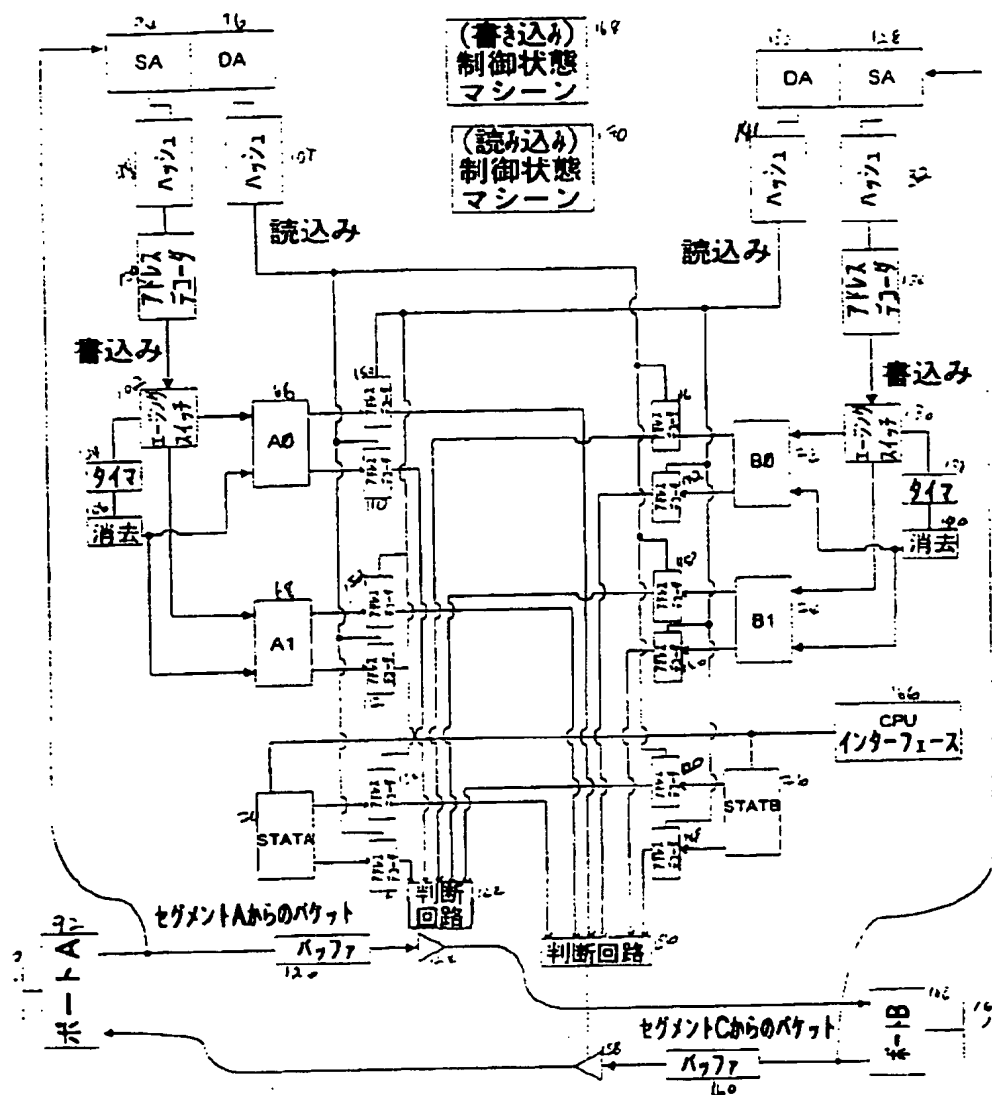
【図 10 b】



【図 20 b】

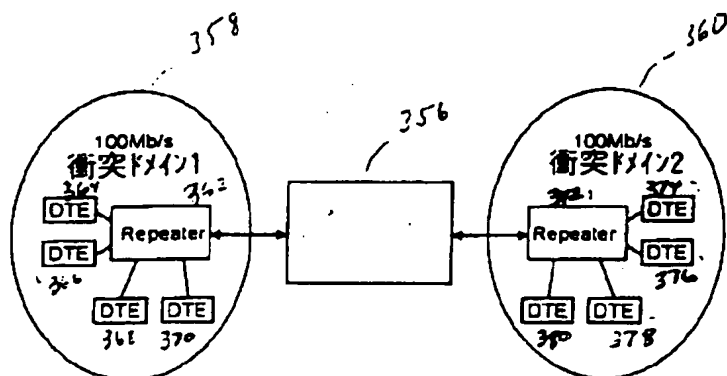
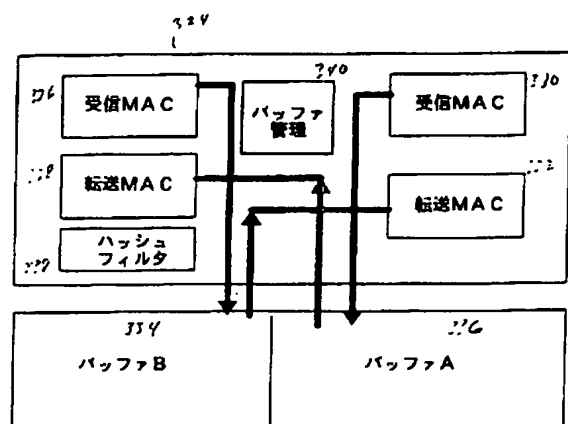


【図4】

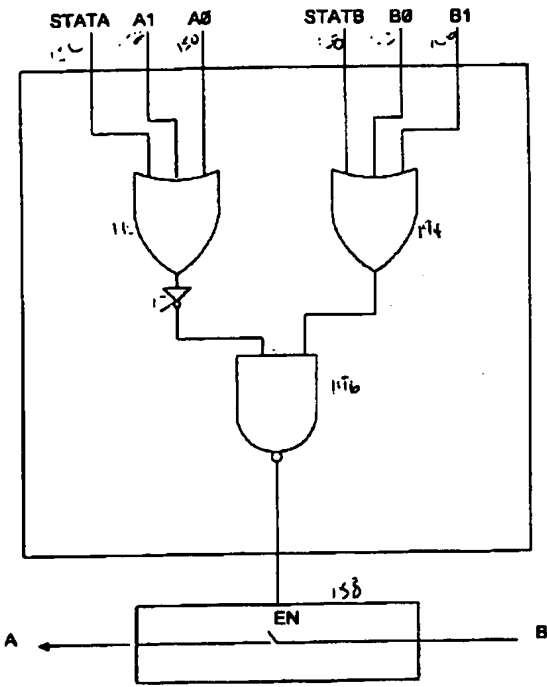


【図16】

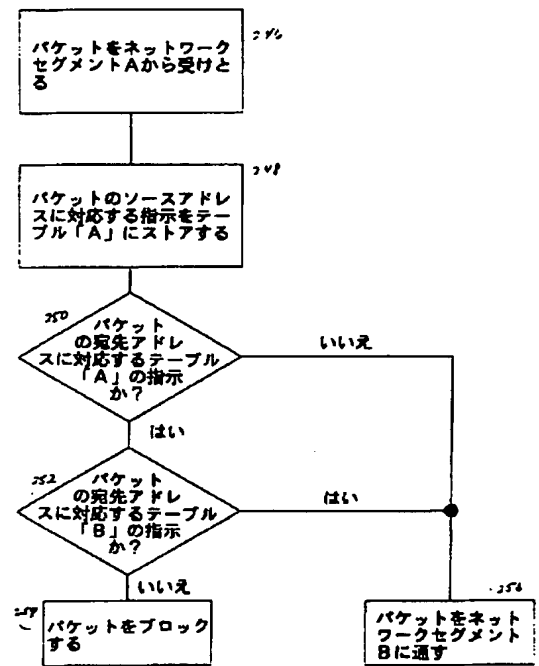
【図18a】



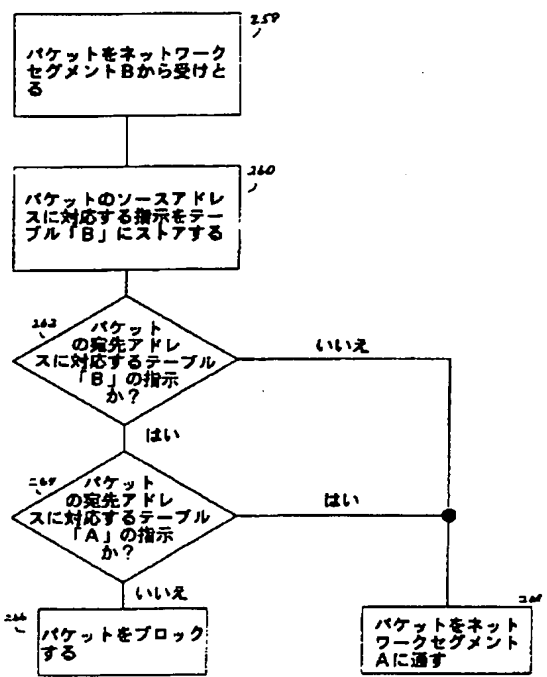
【図 6】



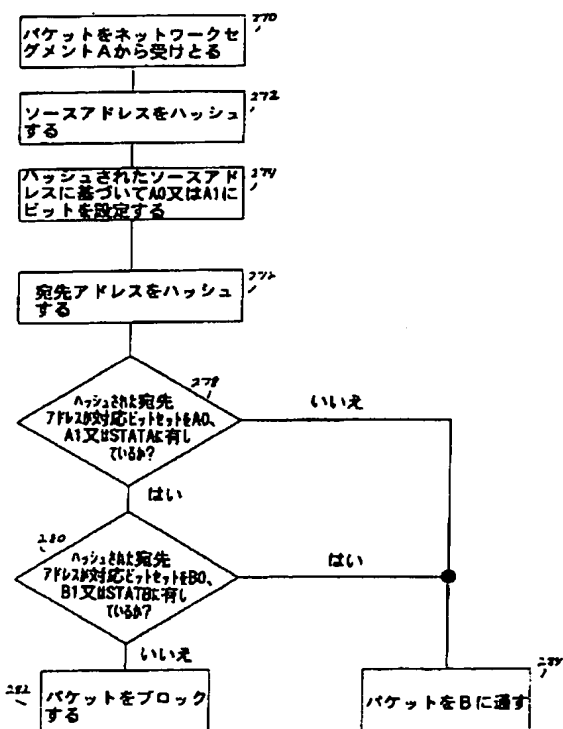
【図 12】



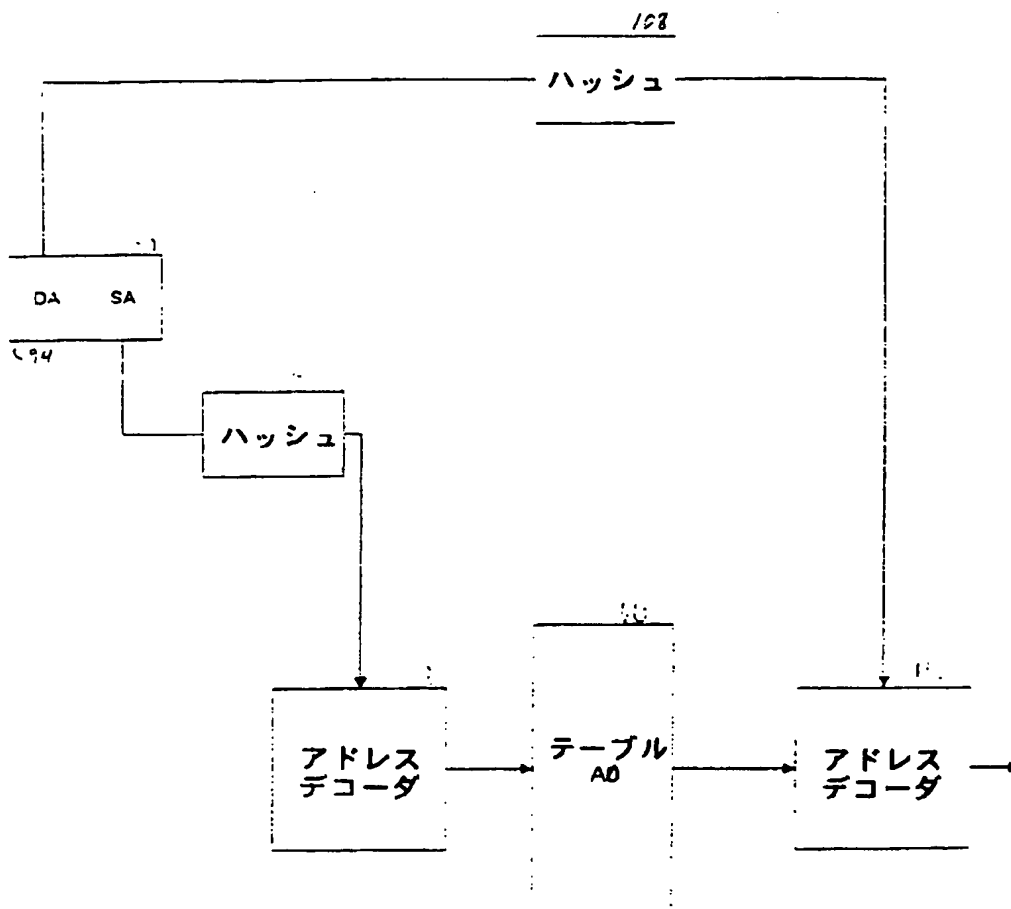
【図 13】



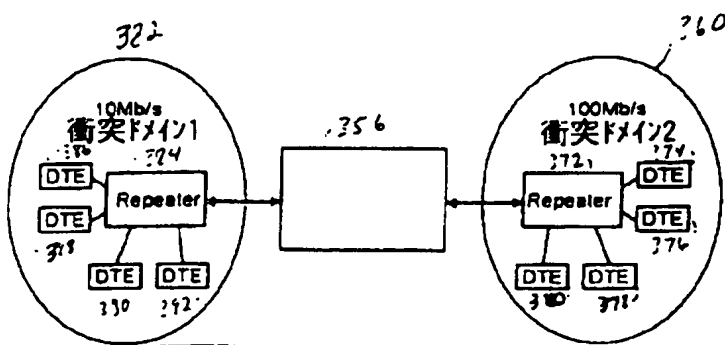
【図 14】



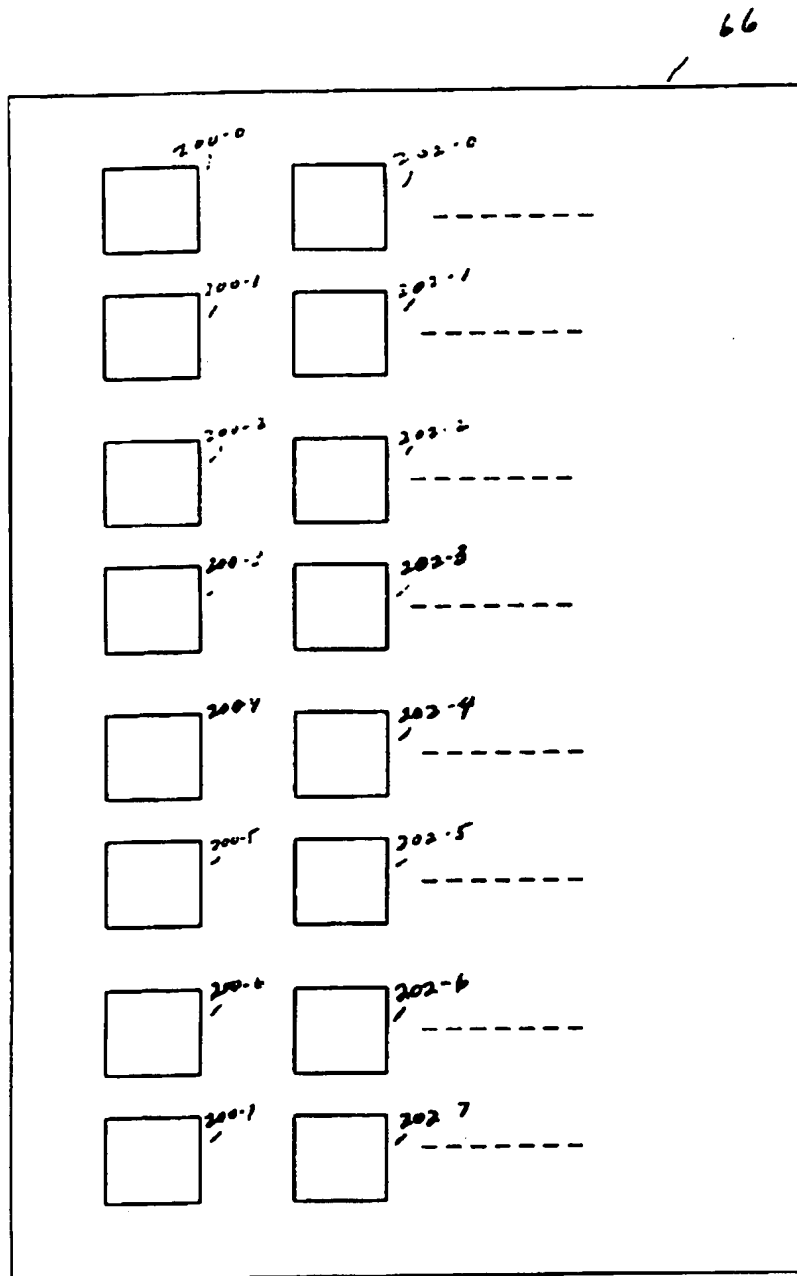
【図 8】



【図 18 b】

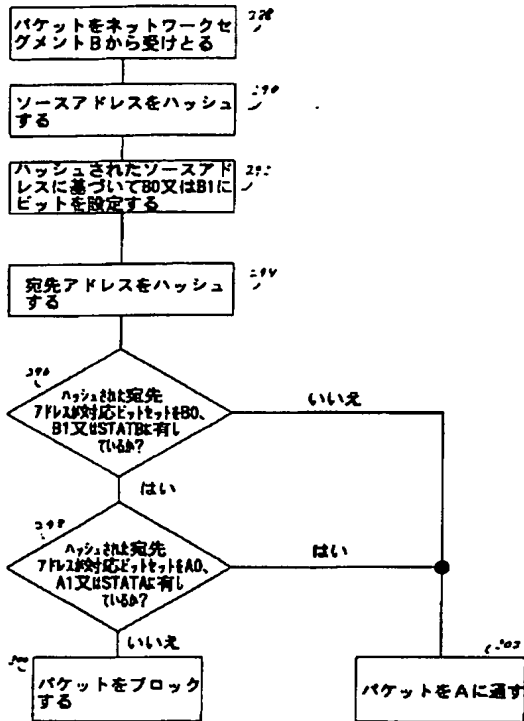


【図 9】

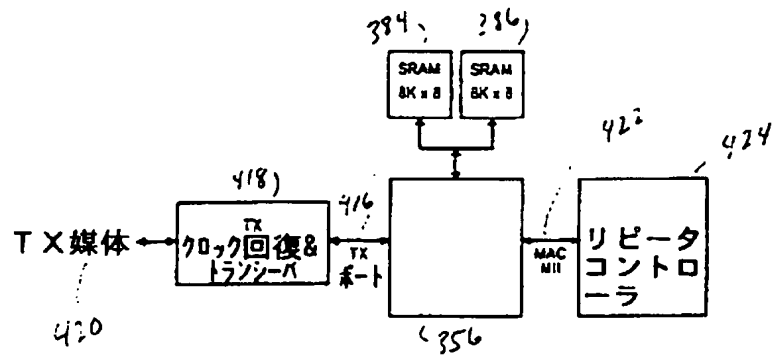




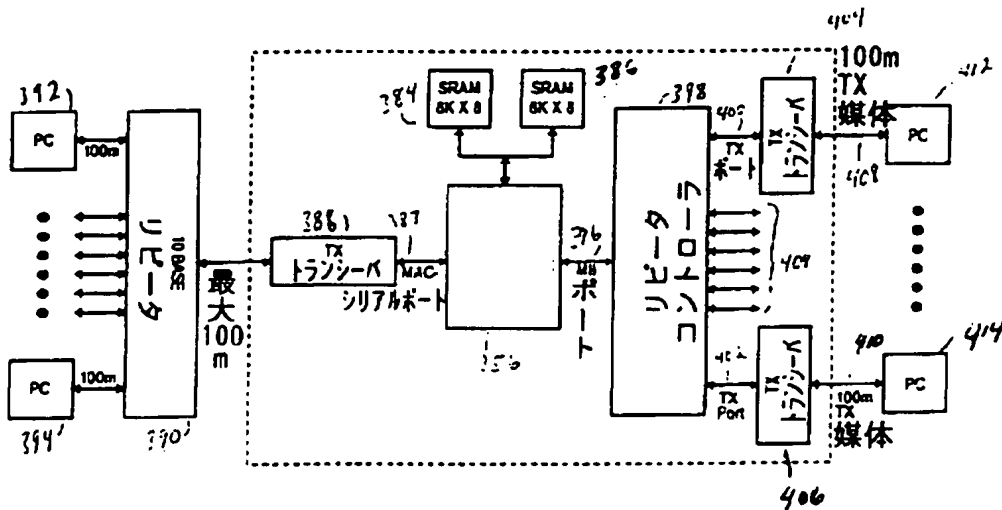
【図15】



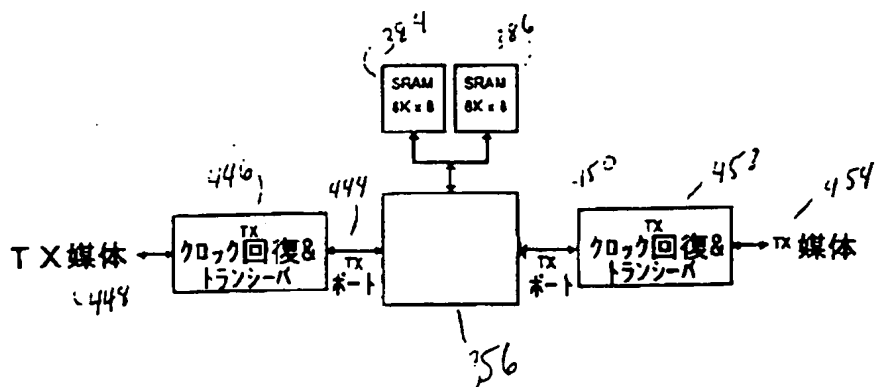
【図20a】



【図19】



【図 20 c】



フロントページの続き

(72) 発明者 ユー リアオ  
 アメリカ合衆国 カリフォルニア州  
 94538 フリーモント レスリー ストリ  
 ート 39663 アpartment 366

(72) 発明者 キース ワー チャウ  
 アメリカ合衆国 カリフォルニア州  
 95131 サン ホセ トアノ コート  
 1520

【外国語明細書】

**BRIDGE DEVICE WITH SELF LEARNING BETWEEN NETWORK  
MEDIA AND INTEGRATED CIRCUIT AND METHOD BASED ON  
THE SAME****BACKGROUND OF THE INVENTION****Field of the Invention**

The present invention relates to the field of devices for interconnecting networks; and more particularly to bridge devices with self learning for interconnecting networks.

**Description of Related Art**

Increasingly, local area networks (LANs) are being used in computing applications. One type of LAN protocol is the Ethernet protocol. In the Ethernet protocol, a network is divided into various segments or "collision domains." Each network segment consists of multiple nodes. A node is a device on a network such as a terminal or a printer. The distance at which nodes in a segment can be located from other nodes is limited. For example, in "Fast Ethernet" within a segment may allow nodes to only be as far apart as 205 meters.

Two segments can be connected through a bridge device. The bridge isolates a collision domain so that a collision that occurs in one segment does not affect another segment. Once the collision domains of two segments are separated by a bridge, two nodes can be farther apart than they could have been had they been in the same collision domain.

Besides segmentation, another bridge function is filtering. The purpose of filtering is to block unwanted traffic of information from entering another segment and wasting bandwidth. A bridge passes information from one segment

to another segment in the form of packets. Based on various criteria, a bridge does not forward some of the packets it receives. For example, a bridge and method for accessing data in a table and its application to the routing of data between remote stations is described in a patent application by Marshall, European Publication Number 0,365,337,A2, (application number 893 (0789.6) published April 25, 1990. For example, inter-LAN connection equipment is described in U.S. Patent No. 5,477,547 (Sugiyama).

One criterion is that if a packet is headed for another node within the same segment (local traffic), preferably the bridge should not forward the packet into another segment. To do this the bridge needs information as to whether particular nodes reside in particular segments. One way a bridge can obtain this information is by observing the source addresses of packets from each segment and storing the source addresses from packets from one segment in a source address table. When a packet arrives, a CPU has to search a table of many entries and compare the destination address with each of them. Alternatively, also using a CPU, the search and compare function may be performed with content addressable memory (CAM). Using a CAM requires extra hardware (the CAM) and is difficult or impossible to scale for networks having gigabyte per second speeds. An FDDI Bridge Frame Learning and Filtering Apparatus and Method where a source address is stored in a CAM is described in U.S. Patent 5,481,540 (Gang).

Accordingly there is a need for efficient and lower cost searching in a bridge to determine whether a packet should be forwarded to another network segment.

If a bridge learns the location of nodes by storing information in tables, then the tables may no longer be accurate if nodes are subsequently moved. There is therefore a need for an apparatus and method to update the table in a bridge so that the tables reflect a more recent view of the location of nodes.

A table with information regarding source addresses may be indexed by means of a hash function. A hash function may be employed to index entries in a table for addresses. However, a problem with hash functions is that two

addresses may map into the same hash value. For example, A Bridge Apparatus with an Address Check Circuit for Interconnecting Networks in which hashing is used is described in U.S. Patent 5,247,620 (Fukuzawa).

Accordingly, there is a need for a device that helps to reduce the complexity and costs associated with a bridge and helps to more efficiently provide the functionality of interconnection between network media and helps to avoid leaks of local packets to other network segments and is scalable for higher speed networks.

### SUMMARY OF THE INVENTION

The present invention provides a device for coupling a first network medium to a second network medium. A first port is coupled to the first network medium and a second port coupled to the second network medium. A memory stores a first plurality of indications and a second plurality of indications. The indications in the first plurality of indications correspond to respective sets of addresses and indicate whether at least one address in the respective set of addresses may be accessible through the first network medium. The indications in the second plurality of indications correspond to respective sets of addresses and indicate whether at least one address in the respective set of addresses may accessible through the second network medium. A connecting circuit is coupled to the first port, the second port, and the memory. The connecting circuit causes the ports to pass or block a packet from the first network to the second network. The packet has a destination address. The connecting circuit causes the ports to pass or block based on a first indication from the first plurality of indications and a second indication from the second plurality of indications. The first indication corresponds to a set of addresses including the destination address of the packet. The second indication corresponds to the set of addresses including the destination address of the packet.

An embodiment of the present invention includes a first circuit that reads a source address of the packet. The first circuit sets a third indication in the first plurality of indications, the indication corresponding to the source address of the packet. A second circuit reads a source address of a second packet from the second port. The second circuit sets a fourth indication in the second plurality of indications. The fourth indication corresponds to the source address of the second packet.

An embodiment of the present invention includes a central processing unit (CPU) interface coupled to the memory that allows a CPU to set indications in the memory.

An embodiment of the present invention includes an erase circuit coupled to the memory that erases old entries from the memory.

An embodiment of the present invention includes an aging timer and an erase circuit coupled to the aging timer and the memory. In this embodiment of the invention the first plurality of indications comprises a third plurality of indications and a fourth plurality of indications. The first circuit is configured to set the third indication in the third plurality of indications or in the fourth plurality of indications based on the aging timer. The erase circuit is configured to erase the third plurality of indications or the fourth plurality of indications based on the aging timer.

In an embodiment of the invention the connecting circuit is configured to block the packet from the first network segment to the second network segment if:

the first indication indicates that at least one address in the second set of addresses may be accessible through the first network medium; and

the second indication does not indicate that at least one address in the fourth set of addresses may be accessible through the second network medium.

In an alternative embodiment of the present invention the connecting circuit is configured to pass the packet from the first network segment to the

second network segment if the second indication indicates that the destination address of the packet may be accessible through the second network segment.

An alternative embodiment of the present invention includes an index generation circuit that generates a first index in response to the destination address of the packet and a selector circuit that selects the first indication from the first plurality of indications based on the first index. The selector circuit is coupled to the connecting circuit and to the index generation circuit. In an alternative embodiment of the invention the index generation circuit comprises a hash circuit and the first index comprises a result of a hash function of the destination address of the packet.

In an alternative embodiment of the invention the respective sets of addresses comprise medium access control (MAC) addresses.

In an alternative embodiment of the invention the first network medium comprises a network using carrier sense multiple access protocol. In another embodiment of the invention, the first network medium comprises a collision domain network segment.

An alternative embodiment is a medium access control (MAC) layer device for coupling a first network medium to a second network medium. A first port is coupled to the first network medium. A second port coupled to the second network medium. A memory has a first table and a second table. A first circuitry monitor first packets from the first port and sets entries in the first table in response to the first packets. The entries in the first table are indexed based on source addresses of the first packets from the first port. At least one entry in the first table maps to more than one address. A second circuitry monitors second packets from the second port and sets entries in the second table in response to the second packets. The entries in the second table are indexed based on source addresses of the second packets from the second port. A third circuitry selects a first entry from the first table and a second entry from the second table based on a destination address of a first packet from the first port. The source addresses of first packets from the first port comprise MAC addresses, and the source addresses of second packets from the second port

comprise MAC addresses. The third circuitry passes the first packet from the first port to the second port if:

the first entry is not set, or

both the first entry and the second entry are set.

The invention helps to provide a cost-effective solution to link fast Ethernet repeaters together so that the distance between nodes can be expanded beyond the collision domain limitation. Flexibility is achieved by designing collision domain networks that are joined by bridges. Self-learning helps to eliminate the need for programming the device. A learning and filtering method helps to avoid the need for specialized CAM hardware.

Other aspects and advantages to the present invention can be seen upon review of the figures, the detailed description and the claims which follow.



### BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 is a schematic block diagram of network segments and a bridge device of the present invention.

Fig. 2 is a schematic block diagram of a bridge device showing blocks for passing packets according to the present invention.

Fig. 3 is a schematic block diagram of a bridge device showing blocks for learning according to the present invention.

Fig. 4 is a more detailed block diagram of a bridge device according to the present invention.

Fig. 5 is a circuit diagram of a decision circuit and a pass through circuit for packets from network A according to the present invention.

Fig. 6 is a circuit diagram of a decision circuit and a pass through circuit for packets from network B according to the present invention.

Fig. 7 illustrates pass through rules according to the present invention.

Fig. 8 is a block diagram of an entry store and select circuit with a table according to the present invention.

Fig. 9 is a block diagram of a table according to the present invention.

Fig. 10a is a block diagram of a hashing function.

Fig. 10b is a block diagram of a cyclic redundancy check circuit.

Fig. 11 is a chart illustrating filtering functions according to the present invention.

Fig. 12 is a flow chart illustrating self learning and filtering for packets from network A according to the present invention.

Fig. 13 is a flow chart illustrating self learning and filtering for packets from network B according to the present invention.

Fig. 14 is a more detailed flow chart illustrating self learning and filtering for packets from network A according to the present invention.

Fig. 15 is a more detailed flow chart illustrating self learning and filtering for packets from network B according to the present invention.

Fig. 16 is a schematic block diagram of a bridge device with medium access control (MAC) blocks according to the present invention.

Fig. 17 is a schematic diagram of a buffer according to the present invention.

Fig. 18a is a schematic block diagram of a bridge device and two 100Mb/s collision domains according to the present invention.

Fig. 18b is a schematic block diagram of a bridge device and one 10Mb/s collision domain and one 100Mb/s collision domain according to the present invention.

Fig. 19 is a schematic block diagram of a combined 100Base-T system according to the present invention.

Fig. 20a is a schematic block diagram of a built-in bridge application according to the present invention.

Fig. 20b is a schematic block diagram of a media independent interface (MII) repeater application according to the present invention.

Fig. 20c is a schematic block diagram of a stand-alone bridge application according to the present invention.

## DETAILED DESCRIPTION

A detailed description of the preferred embodiments of the present invention is provided with reference to the figures.

Figure 1 illustrates the architecture of an embodiment of the present invention and a context for use of the present invention. Segments 20, 24, 26, 28, and 30 are interconnected via bridge device 21. Bridge device 21 receives information from the segments 20, 24, 26, 28, and 30 and determines whether to pass the information to other segments. Packets are sent to local addresses (within the segment from which the packet originated) or to non-local addresses (outside the segment from which the packet originated). Bridge 21 helps to block packets that are destined to local addresses and pass packets that are destined to non-local addresses. Filter 44 selectively blocks or passes packets from one network segment to another.

By observing source addresses of packets received on various ports 32, 34, 36, 38, 40, and 42, bridge device 21 learns which network segments are associated with the addresses. When bridge device 21 receives a new packet, bridge device 21 updates its knowledge of the location of the address associated with the source address of the packet and also decides whether to forward the packet based on the destination address of the packet.

Figure 1 shows a bridge which interconnects a total of six network segments (20, 24, 28, 22, 26 and 30). An alternative embodiment bridge 21 could be implemented to interconnect another number of network segments. For example, bridge 21 could be designed to interconnect a total of two network segments. In one preferred embodiment, bridge 21 includes a simple integrated circuit including control circuitry for connecting two network segments.

Figure 2 is a schematic block diagram of a bridge device showing blocks for passing packets according to the present invention. Figure 2 includes a network segment A 53 including network nodes 46, 48, 50 and 52, a second network segment B 55 including network nodes 54, 56, 58, 60 and 62, and a bridge device 45 interconnecting network segments 53 and 55. Bridge device 45

includes filter 64 and a memory 65 comprising dynamic table A0 66, dynamic table A1 68, static table A 70, dynamic table B0 72, dynamic table B1 74, static table B 76. Bridge device 45 is coupled to network segment 53 and network segment 55.

Filter 64 determines whether to pass packets from network segment A 53 to network segment B 55 based on information contained in dynamic table A0 66, dynamic table A1 68, static table A 70, dynamic table B0 72, dynamic table B1 74, and static table B 76. The tables contain indications of whether a node exists on a particular side of the bridge. Since the indications are indexed based on hash value, there is the possibility that two different addresses may map to the same hash value. Because of this possibility, if a positive indication is found on a side A table corresponding to an address of a packet from network segment A, the filter will also check side B tables.

Figure 3 is a schematic block diagram of a of a bridge device 45 for learning according to the present invention. Figure 3 shows a packet from segment A, including a source address 78 and a destination address 80 and a packet from segment B, including a source address 82 and destination address 84. Bridge 45 includes hash circuit 86, dynamic table A0 66, dynamic table A1 68, static table A 70, hash circuit 88, dynamic table B0 72, dynamic table B1 74, and static table B 76.

When bridge device 45 learns addresses, indications corresponding to source addresses of packets are stored in tables associated with the network segment from which the packets originated. Source address 78 is hashed by hash circuit 86 to create a hash value, and an indication is stored indexed by the hash value in either table A0 66 or table A1 68. Source address 82 from segment B 55 is hashed by hash circuit 88 to create a hash value, and an indication is stored in either table B0 72 or B1 74, indexed by the hash value. The use of multiple dynamic tables on each side of bridge 45 allows for the aging and flushing of old entries of the table. For example, new indications may be stored in dynamic table A0 66, while old indications may have been stored in dynamic table A1 68. After a period of time (e.g., 5 minutes), old entries from

table A1 68 are erased and then dynamic table A0 68 becomes the old table and new entries are written into dynamic table A1 68. Similarly, dynamic table B0 72 and dynamic table B1 74 are used in conjunction in order to allow for the aging of indications and for the erasing of old indications. Static table A 70 and static table B 76 are written by a CPU in order to provide indications of whether nodes exist on a particular side of the bridge when those network nodes do not normally broadcast their presence so that their presence may not be recorded in the dynamic tables.

Figure 4 is a more detailed block diagram of a bridge device according to the present invention. Figure 4 includes a source address 94 from segment A 90, a destination address 96 from segment A 90, a source address 128 from segment B 164, a destination address 130 from segment B 164, and supporting circuitry for filtering packets originating from segment A 90 or from segment B 164.

Port 92 is coupled to network segment A 90 and receives packets from segment A 90. Port 92 is coupled to buffer 126 for storing packets. Port 92 provides a source address 94 to hash circuit 98. Source address 94, destination address 96, destination address 130, and source address 128 are each 6 byte MAC level addresses. Hash circuit 98 provides an 8-bit hash value in response to the 6 byte source address 94. Hash circuit 98 is coupled to address decoder 100. Address decoder 100 is coupled to dynamic table A0 66 and dynamic table A1 68 through aging switch 102. Dynamic tables A0 66 and dynamic tables A1 68 store indications indexed by the hash values generated by hash circuit 98 to indicate that source addresses corresponding to the indications exist on network segment A 90. Aging switch 102 is coupled to timer 104 and controls in which table A0 66 or table A1 68 indications are stored. Erase circuit 106 is coupled to dynamic table A0 66 and dynamic table A1 68 and erases old entries in the dynamic tables based on timer 104.

Hash circuit 108 is coupled to port 92 and receives a destination address 96 from a packet from segment A 90 and hashes the destination address to form a hash value. The hash value from hash circuit 108 is used to index into hash tables and obtain indications corresponding to the hash value. Hash circuit 108

is coupled to address decoder 110, address decoder 112, address decoder 114, address decoder 116, address decoder 118, and address decoder 120. Address decoder 110 is coupled to dynamic table A0 66. Address decoder 112 is coupled to dynamic table A1 68. Address decoder 114 is coupled to static table A 70. Address decoder 116 is coupled to dynamic table B0 72. Address decoder 118 is coupled to dynamic table 174. Address decoder 120 is coupled to static table B 76. The inputs of decision circuit 122 are coupled to address decoder 110, address decoder 112, address decoder 114, address decoder 116, address decoder 118, and address decoder 120. The output of decision circuit 122 is coupled to the control of pass through switch 124. The input of pass through switch 124 is coupled to buffer 126. The output of pass through switch 124 is coupled to port 162, which is coupled to segment B 164.

Segment B 164 is coupled to port B 162. Hash circuit 132 receives a source address 128 from port 162. Hash circuit 132 is coupled to address decoder 134 and provides a hash value to address decoder 134. Address decoder 134 is coupled through aging switch 136 to dynamic table B0 72 and dynamic table B1 74 for writing indications into table B0 72 or table B1 74 indexed by the hash value from address decoder 134. Aging switch 136 is coupled to timer 138 and controls into which table indications are written depending on timer 138. Erase circuit 140 is coupled to dynamic table B0 72 and dynamic table B1 74 and erases old entries in respective tables. CPU interface 166 is coupled to a CPU and to static table A 70 and static table B 76 and allows a CPU to write entries into the static tables.

Hash circuit 141 is coupled to port 162 to receive destination address 130 and to provide a hash value in response to destination address 130. Hash circuit 141 is coupled to address decoder 142, address decoder 146, address decoder 148, address decoder 150, address decoder 152, and address decoder 154. Address decoder 142 is coupled to dynamic table B0 72. Address decoder 146 is coupled to dynamic table B1 74. Address decoder 148 is coupled to static table B 76. Address decoder 150 is coupled to dynamic table A0 66. Address decoder 152 is coupled to dynamic table A1 68. Address decoder 154 is

coupled to static table A 70. The input of decision circuit 156 is coupled to address decoder 142, address decoder 146, address decoder 148, address decoder 150, address decoder 152, and address decoder 154. The output of decision circuit 156 is coupled to the control input of pass through circuit 158. Pass through circuit 158 is coupled to buffer 160 and to port 92.

Write control state machine 168 controls the operation of the bridge during the writing of indications corresponding to source addresses. Read control state machine 170 controls the reading of indications and the passing or blocking of packets from network segments.

When a new packet from segment A 90 arrives at port 92, the source address of the packet 94 is hashed to create an 8-bit hash value. A 1-bit indication is stored in one of the dynamic tables (dynamic table A0 66 or dynamic table A1 68). This 1-bit indication is indexed by the hash value that was obtained by hashing the source address of the packet. Each table contains 256 1-bit entries in order to provide entries corresponding to 256 possible hash values. Table A0 66 or table A1 68 are written alternatively as controlled by aging switch 102 depending on timer 104. In this manner, dynamic table A0 66 and dynamic table A1 68 alternate in the roles of active and historic tables. When the timer expires after 5 minutes, the table having the older entries (historic table) is erased by erase circuit 106. Then the historic table becomes the active table and entries are written into it for the next five minutes.

Similarly, the source address of a packet from network B is hashed to obtain a hash value and an indication (1 bit) is stored in either dynamic table B0 72 or dynamic table B1 74 to indicate that an address corresponding to the source address is present on side B. Also, dynamic table B0 72 and dynamic table B1 74 are used alternately for writing indications in order to provide for aging and erasing of old indications. Static table A 70 and static table B 76 provide for storing of indications corresponding to nodes that do not normally broadcast packets and thus ordinarily would not have corresponding indications in the dynamic tables. A CPU writes indications in static table A 70 and static table B 76 via a CPU interface 166. Decision circuit 122 controls whether

packets pass from segment A 90 to segment B 164. Whether a packet is passed from segment A to segment B is determined by decision circuit 122 based on the indications stored in static tables and the dynamic tables. Similarly, whether a packet is passed from segment B to segment A is determined by decision circuit 156 based on indications in the static tables and the dynamic tables.

When a packet arrives from a network segment, indications are stored in the corresponding dynamic tables to indicate the source of that packet. The destination address of the packet is used to determine whether the packet should be forwarded to another network segment based on the indications in tables corresponding to the destination address.

Figure 5 is a circuit diagram of a decision circuit and a pass through circuit for packets from network A, according to the present invention. The input of OR gate 184 is coupled to static table A 70 via address decoder 114, to dynamic table A1 68 via address decoder 112, and to dynamic table A0 66 via address decoder 110. The input of OR gate 180 is coupled to static table B 76 via address decoder 120, to dynamic table B0 72 via address decoder 116, and to dynamic table B1 74 via address decoder 118. The output of OR gate 184 is coupled to the input of NAND gate 186. The output of OR gate 180 is coupled to the input of inverter 182. The output of inverter 182 is coupled to the input of NAND gate 186. The output of NAND gate 186 is coupled to the control of pass through circuit 124 such that pass through circuit will allow a packet to pass if an indication corresponding to the packet has been set in static table B 76, dynamic table B0 72, or dynamic table B1 74 and will allow a packet to pass if no corresponding indication is set in static table A 70, dynamic table A1 68, and dynamic table A0 66.

Figure 6 is a circuit diagram of a decision circuit and pass through circuit for packets from network B, according to the present invention. The input of OR gate 190 is coupled to static table A 70 via address decoder 154, to dynamic table A1 68 via address decoder 152, and to dynamic table A0 66 via address decoder 150. The input of OR gate 194 is coupled to static table B 76 via address decoder 148, to dynamic table B0 74 via address decoder 142, and to



dynamic table B1 74 via address decoder 146. The output of OR gate 190 is coupled to the input of inverter 192. The output of OR gate 194 is coupled to the input of NAND gate 196. The output of NAND gate 196 is coupled to the control of pass through circuit 158. Decision circuit 156 controls whether a packet will pass network segment B 164 to network segment A 90.

Figure 7 illustrates the pass through rules according to the present invention. The number 0 represents that an indication has not been set for a particular table, for example, the first entry under A0 is 0. This represents that the corresponding indication is not currently set. The number 1 indicates that the corresponding indication is set. For example, the second entry under A0 is 1 and indicates that a corresponding entry of A0 is set. The letter "x" indicates that for that particular row that corresponding value signified with the "x" is irrelevant, and whether it is 0 or 1, the result will be the same. In the first row the chart shows pass through enabled, (as represented by 1 in the enable column EN) when the corresponding entry of A0 has not been set, A1 has not been set and static table A has not been set. For that particular state represented in the first row of the chart, the values of B0, B1, and static table B are irrelevant as represented by x's in the corresponding entries. Thus, a packet is passed from segment A to segment B if none of the tables corresponding to segment A nodes have been set with corresponding indications, and a packet is passed from segment A to segment B if any of the tables corresponding to segment B have been set with corresponding indications. The second half of Figure 7 shows the rules for passing packets from segment B 164 to segment A 190.

Figure 8 is a block diagram of an entry store and select circuit with a table according to the present invention. Address decoder 100 and address decoder 110 are each implemented as multiplexers. Hash circuit 98 hashes source address 96 to create a first hash value. The first hash value is used by address decoder 100 to write an entry in dynamic table A0 66. The entry that is written in dynamic table 66 is indexed by the first hash value provided by hash circuit 98. Destination address 94 is hashed by hash circuit 108 to create a second hash value. The second hash value is used by address decoder 110 in

selecting an indication corresponding to the destination address 94 from dynamic table A0 66. Since a 6-byte address (MAC address) is used to hash into an 8-bit hash value, there is the possibility that a set of two or more addresses will hash to the same hash value. Thus, if the second hash value has been set, it may have been set in response to a packet having a source address equal to destination address 94, or it may have been set by a packet having a source address that is not equal to destination address 94 but hashes to the same 8-bit hash value.

Figure 9 is a block diagram of a table according to the present invention. Dynamic table A0 66 is constructed in a similar manner to other tables (dynamic table A1 68, dynamic table B0 72, dynamic table B1 74, static table A 70, and static table B 76) used by the bridge device. Table A0 66 is comprised of a total of 256 entries, each entry corresponding to an indication of whether at least one address in a set of addresses exists on a particular side of the bridge. Each indication among the 256 indications is stored as a 1-bit value. Each 1-bit value is stored by a single D flip flop, for example, flip flop 200-0. Thus, dynamic table A0 is comprised of flip flop 200-0 through flip flop 200-7, flip flop 202-0 through flip flop 202-7, and other flip flops comprising a total of 256 flip flops. Indications are written into table A0 66 by setting a flip flop at a location in table A0 66 indexed by a hash value generated by hash circuit 98 and as selected in table A0 66 by address decoder 100. Table A0 is erased by clearing all flip flops comprising table A0. The table is comprised of 32 8-bit registers. Each column shown in Figure 9 comprises an 8-bit register, for example, the row comprised of flip flop 200-0 through flip flop 200-7. After reset, all tables are cleared to "0's" which means that all packets received on corresponding sides are forwarded. Static table entries are set and cleared by an external CPU. Dynamic tables are set and cleared during the learning and aging process.

Fig. 10a is a block diagram of a hashing function according to an embodiment of the present invention. The first six bytes of a destination address 210 of a non-multicast and non-broadcast packet are fed through the IEEE standard 802.3 Ethernet cyclic redundancy check (CRC) function (block 212). A 32-bit CRC value 214 results from the CRC function 212. The most

significant 8 bits of the 32-bit CRC value 214 are latched (block 216). These 8 bits are decoded to index an indication (1 bit) in the table (e.g., dynamic table A0 66, dynamic table A1 68, dynamic table B0 72, or dynamic table B1 74). To decode, the least significant 5 bits (item 219) index into one register among the 32 registers of the table (e.g., dynamic table A0 216), and the most significant 3 bits (item 220) index to a bit in the selected register. In the example shown in figure 10a, the destination address is found to hash to register 21, bit 2. The hashing function is also used for hashing a source addresses to store indications stored in the tables and for hashing of destination addresses in order to obtain indications from the static tables (static table A 70 and static table B 76).

Figure 10b is a block diagram of a cyclic redundancy check (CRC) circuit. Address bits are obtained from media independent interface (MII) 460 and are processed by the CRC circuit to yield an 8-bit hash value 482. CRC circuit included register 462 and 464, exclusive OR logic (XOR) blocks 466, 476, 478, and 480, and D flip-flops 468, 470, 472, and 474.

Address bits from MII 460 are received by register 462. The output of register 462 is an 4-bit value and is coupled to the input of register 464 and to the input of XOR 466. The output of register 464 is a 4-bit value and is combined with the output from register 462 as an 8-bit input to XOR 466. The output of XOR 466 is coupled to the input of XOR 476, to the input of XOR 478, to the input of XOR 480, and to the input of D flip-flop 474. The output of XOR 476 is coupled to the input of D flip-flop 468. The output of D flip-flop 468 is coupled to the input of XOR 466 and as an 8-bit hash value 482. The output of XOR 478 is coupled to the input of D flip-flop 470. The output of D flip-flop 470 is coupled to input the input of XOR 476. The output of XOR 480 is coupled to the input of D flip-flop 472. The output of D flip-flop 472 is coupled to the input of XOR 478. The output of D flip-flop 474 is coupled to the input of XOR 480.

Although a CRC hashing function is described here, it is appreciated that other deterministic randomizing functions could be used to create index values by which to index indications in the tables. For a discussion of a cyclic

redundancy check function, see An American Standard IEEE Standards for Local Area Networks: *ANSI/IEEE Std 802.3-1985 ISO Draft International Standard 8802/3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, Technical Committee on Computer Communications of the IEEE Computer Society (Approved June 24, 1983, IEEE Standards Board; December 21, 1984 American National Standards Institute), which is incorporated herein by reference. See, in particular, section 3.2.8 ("Frame Check Sequence Field") of that document.

Figure 11 is a chart illustrating filtering functions according to the present invention. In the cases of 1 through 16 where self-learning mode is selected, the number of packets that are blocked include those that are filtered by each individual case as well as those rejected by the self-learning tables. If an inverse filtering option is activated (case 17, 18, 19), only broadcast or multicast packet is forwarded and the rest of the packets are blocked regardless of the setting of filtering options. Broadcast Packet Filter: Packets with all "1's" in the 48-bit Destination Address are not forwarded. Multicast Packet filter: Packets with "1" in the Group Address bit are not forwarded; this does not include broadcast packets. Self-Addressing Packet (DA=SA) Filter: Packets with the same Destination and Source Addresses are not forwarded. Inverse Broadcast Filter: Only packets with all "1's" in the 48-bit Destination Address are forwarded. Inverse Multicast Packet Filter: Only packets with "1's" in the Group Address but not Broadcast address are forwarded.

256-bit Static Hash Filter: There are two static hash filter tables STATA and STATB. The destination address of a non-multicast/broadcast packet from Port A is hashed as described above to provide the corresponding bits in STATA and STATB. If the indexed bit in STATA is set to "1" and the corresponding bit in STATB is not set to "1", this packet is blocked from transmitting into Port B. If the indexed bit in STATA and the corresponding bit STATB are both set to "1", the packet is not blocked. Hash filtering from Port B to Port A works in the same way.

Figure 12 is a flow chart illustrating self-learning and filtering for packets from network A according to the present invention. First a packet is received from segment A (step 246). In table A an indication corresponding to the source address of the packet is stored. Next table A is checked to determine whether an indication in table A exists corresponding to the destination address of the packet (step 250). If an indication in table A corresponding to the destination address to the packet does not exist, then the packet is passed to network segment B (step 256). If an indication in table A corresponding to the destination address of the packet does exist, then it is determined whether an indication in table B corresponding to the destination address of the packet exists (step 252). If an indication in table B corresponding to the destination address of the packet does exist, then the packet is passed to network segment B (step 256). If an indication in table B corresponding to the destination address of the packet does not exist, then the packet is blocked (step 254). Thus, if an indication in table A corresponding to the destination address of the packet does not exist, then table B is not checked. Otherwise table B will be checked. Table B is checked if an indication in table A corresponding to the destination address of the packet does exist in order to help account for the situation where another address in the set of addresses hashing to the same hash value has caused the indication in table A to be set.

Figure 13 is a flow chart illustrating self-learning and filtering for packets from network B according to the present invention. A packet is received from network segment B (step 258). An indication is stored in table A corresponding to the source address of the packet (step 260). Table B is checked to determine whether an indication corresponding to the destination address of a packet exists in table B (step 262). If an indication in table B corresponding to the destination address of the packet does not exist, then the packet is passed to network segment A (step 268). If an indication in table B corresponding to the destination address of the packet does exist, then the table A is checked to determine whether an indication corresponding to the destination address of the packet exists in table A (step 264). If an indication in table A corresponding to

the destination address of the packet exists, then the packet is passed to network segment A (step 268). If an indication in table A corresponding to the destination address of the packet does not exist, the packet is blocked (step 266).

Figure 14 is a more detailed flow chart illustrating self learning and filtering for packets from network segment A according to the present invention. A packet is received from side A (step 270). The source address of the packet from side A is hashed (step 272) to form a source hash value. A bit is set in table A0 or table A1 based on the source hash value (step 274). Next, the destination address is hashed to form a destination hash value (step 276). Next, tables A0, A1, and stat A are checked to determine whether any of these tables has a indication bit set corresponding to the destination hash value (step 278). If none of these tables has a bit set corresponding to the destination hash value, then the packet is passed to segment B (step 284). If at least one of the tables A0, A1, or stat A has a bit set at an entry indexed by the destination hash value, then the tables B0, B1, and stat B are checked at locations corresponding to the destination hash value (step 280). If at least one corresponding bit indexed by the destination hash value is set in table B0, B1, or stat B, then the packet is passed to segment B (step 284). If no bit set indexed by the destination hash value in either B0, B1, or stat B, then the packet is blocked (step 282).

In step 274 a bit is set in either table A0 or table A1 depending on an aging timer. In this manner, table A0 and A1 allow for a set of older indications and a newer set of indications to be stored. After a determined amount of time (5 minutes), entries in the older table are erased and the newer table then becomes the older table and then new entries will be written into what was previously the older table. An aging timer is used to determine the time interval for maintaining the dynamic hash tables before being flushed. The aging timer interval is selected by adjusting pins or by programming an aging timer register on an integrated circuit embodiment of the invention. In one embodiment of the invention the minimum aging time is 5 minutes, and the maximum is 1275 minutes. Alternatively, the aging timer may be turned off so that no flushing of

old entries occurs. Other embodiments of the invention having other possible aging time settings are possible.

Figure 15 is a more detailed flow chart illustrating self learning and filtering for packets from network B according to the present invention. A packet is received from side B (step 288). The source address of the packet from side B is hashed to form a source hash value (step 290). A bit in table B0 or table B1 is set corresponding to the source hash value obtained in step 290 (step 292). The destination address of the packet is hashed to form a destination hash value (step 294). It is determined whether bits corresponding to the destination hash value exist in either table B0, table B1, or table stat B. If no corresponding indication bits indexed by the destination hash value are set in table B0, table B1, or table stat B, then the packet is passed to segment A (step 302). If a corresponding entry indexed by the destination hash value is present in table B0, table B1, or stat B, then table A0, table A1, and table stat A are checked to determine whether they contain indication bits indexed by the destination hash value (step 298). If a indication is set indexed by the destination hash value in either table A0, table A1, or stat A, then a packet is passed to segment A (step 302). If no indication indexed by the destination hash value is set in table A0, table A1, or stat A, then the packet is blocked (step 300).

Figure 16 is a schematic block diagram of a bridge device with medium access control (MAC) blocks according to the present invention. Bridge device 324 includes receive MAC 326, transmit MAC 328, receive MAC 330, transmit MAC 332, hash filter 338, buffer management block 340, buffer B 334, and buffer A 336. Receive MAC 326 receives packets from a first network and buffers packets in buffer B 334. Transmit MAC 332 transmits packets from buffer B 334 to a second network. Receive MAC 330 receives packets from a second network and buffers those packets into buffer A 336. Transmit MAC 328 transmits packets from buffer A 336 to the first network. Hashed filter 338 controls whether a packet is passed from the first network to the second network or from the second network to the first network based on hash values of destination addresses of the packets and based on tables which store indications

corresponding to addresses of packets. Buffer management 340 manages buffer B 334 and buffer A 336. The MACs are fully IEEE 802.3 compliant in half-duplex and full-duplex implementation.

Figure 17 is a schematic diagram of a buffer according to the present invention. Buffer 344 stores packets from network segments coupled to port A and to port B. Memory is partitioned into two sections. Section A is a receive buffer for port A; section B for port B. Buffer 344 comprises a size of 64 kilobytes. Packets from port A (packet 346, 348, 350, for example) are stored in the first 32 kilobytes of the buffer 344. Packets from network segment B (packet 352, 354, for example) are stored in the second 32 kilobytes of buffer 344. The buffers are implemented with high speed (20 nanoseconds or faster) SRAMs. A minimum of 16 Kbyte buffer memory is required, but 256 Kbyte buffer is preferred for higher performance. Size of each section is configurable through external pins. Alternatively, auto sizing can be used to allow dynamic buffer allocation once a section of a buffer is full. Figure 18 shows the initial configuration of buffer 344 where allocation is one to one between port A and port B. The last word of each buffer is status storage. After receiving a packet, the bridge writes the packet length in the storage. A bad packet is rejected at the end and the buffer is reclaimed. After reset, both sides are selected at 100 Mps, the buffer sizes in A and B are equal. Minimum size of a packet buffer is 2 kbytes. The buffer memory has four modes:

- (1) Receiving from A, transmitting to B,
- (2) Receive from B, transmitting to A,
- (3) Receiving from A, receiving from B,
- (4) Transmitting to A, transmitting to B.

If after the whole packet is received and no more buffer space is available, a buff\_full pin will be asserted for as long as there is no empty buffer. In the full duplex mode, a pause packet is sent to the other end to signal that no more buffer space is available if a flow control mechanism is enabled. The time interval of a pause timer to cause the other end to stop sending packets is determined by a serial EEPROM.



Two possible forwarding modes and two transmission protocols can be selected through a pin on the bridge device. The forwarding modes are either full-packet store and forward, or 64-byte store and forward. In the full-packet store and forward mode, packet is forwarded only after the complete packet is received and checked and an ill-formed packet will be discarded and the buffer reclaimed. In the 64-byte store and forward mode, the bridge is to transmit the receiving packet once the first 64 bytes are received without collision; however, if during the receiving, another packet is being transmitted or waiting to be transmitted, this option will not have the effect, i.e., received packet will be buffered in full. With half-duplex transmission protocol, the carriers sense multiple access with collision detect (CSMA/CD) protocol is implemented. In full-duplex and full control protocol, without collision and a carrier sense detection, transmit starts at the 64th byte location or at the end of a packet. The pause packet is received, transmission activity is suspended until the pause time expires.

Four types of physical interfaces on port A and six types of physical interfaces on Port B to accommodate different applications are provided. These interfaces are 10/100 MII Data Interface, 100M TX/FX Interface, and 10M Serial Interface. The physical interfaces of port A and B can be selected independently through pins corresponding to the respective ports. The most commonly used 10/100 MII Data Interfaces which comprise framed 4-bit-nibble-wide synchronous data path and control signals are categorized in two groups. The MAC-MII Interface accepts data through RXD[3:0] from a PHY device and sending data through TXD[3:0], and the PHY-MII Interface accepts data through TXD[3:0] from a MAC-like or a repeater device and sending data through RXD[3:0]. The 10M MII 4-bit-nibble mode is also available by transferring data at 2.5MHz clock rate. The 100M TX/FX Interface allows transferring information to and from the MAC or a repeater using 5-bit of unframed data with lower latency at 25 MHz clock rate. The 5-bit code-groups are undetectable and have no meaning outside the 100 BASE-X physical protocol data unit, called a "Stream". Only Port B can be configured to 10M

Serial mode to transfer data with a serial data stream at 10 MHz clock rate. Two types of 10M serial modes are available; the MAC-Serial mode allows the “7-wire” interface to a 10Base-T PHY and the PHY-Serial mode allows the “7-wire” interface to a 10Base-T MAC. These interfaces are selected by setting PHY2-0\_B at (1, 0, 0) for MAC-Serial mode and (1, 0, 1) for PHY-Serial mode.

The interface selections are detailed as follows:

<u>Physical Interface on Port A</u>	<u>Speed (Mb/s)</u>
MAC-MII(Connecting to a PHY)	10/100
PHY-MII(Connecting to a MAC)	10/100
FX	100
TX	100

<u>Physical Interface on Port B</u>	<u>Speed (Mb/s)</u>
MAC-MII(Connecting to a PHY)	10/100
PHY-MII(Connecting to a MAC)	10/100
FX	100
TX	100
MAC-Serial(Connecting to a PHY)	10
PHY-Serial(Connecting to a MAC)	10

Three types of MII Interfaces are provided: the 100 Mb/s MII Interface, the 10 Mb/s Nibble Interface, and the 10 Mb/s Serial Interface. The 10/100 Mb/s MII operates in 4-bit nibble mode at 2.5/25 MHz clock rate. To provide more flexibility for various applications, the reversed MII mode for each MII application is also implemented. For instance, the MAC-MII interface is used to connect to a PHY device, but its reversed MII, the PHY-MII Interface, is used to connect to a MAC device. The MAC-MII comprises of 15 signals where RXDV, CRS, COL, RXER, RXD[3:0] are input and TXEN, TXD[3:0] are output. In MAC-MII mode, the clock lines TXCLK and RXCLK sourced by the Physical Layer chip supply the transmit and receive clocks for the bridge device with 2.5 MHz at 10 M speed and 25 MHz clock rate at 100 M speed. On the

contrary, the PHY-MII interfaces to a MAC device with RXDV, CRS, COL, RXER, RXD[3:0] as output and TXEN, TXD[3:0] as input are provided for PHY-MII interface. The TXCLK which provides clock reference for the MX98201 and the Physical Layer chip is sourced by an external 2.5/25 MHz clock; the RXCPHY is sourced by the bridge device. For 10 Mb/s serial Interface operation, the MII Data Interface is remapped to accommodate the 10 Mb/s Interface. The MAC-Serial Interface is used to provide RXD, CRS, COL as input to the bridge device and TXEN, TXD as output to a 10 Mb/s Ethernet Physical Layer. The TXCLK and RXCLK clock lines generated by the Physical Layer chip provide the 10 MHz transmit and receive clocks for the bridge device. On the other hand, the PHY-Serial Inter-face uses RXD, CRS, COL as output to a MAC device and TXEN, TXD as input. An external 10 MHz clock supplies TXCLK for the MX98201 and the MAC device; the RXCPHY is sourced by the bridge device.

Figure 18a is a schematic block diagram of a bridge device and two 100Mb/s collision domains according to the present invention. The design shown in Figure 18a uses a single bandwidth (100Mb/s) in both collision domains. The first collision domain 358 includes a repeater 362, which is coupled to DTE 364, DTE 366, DTE 368, and DTE 370. The second collision domain 360 includes repeater 372, DTE 374, DTE 376, DTE 378, and DTE 380. The first collision domain 358 is coupled to the second collision domain 360 through bridge device 356. Bridge device is coupled to repeater 362 in the first collision domain 358 and to repeater 372 in the second collision domain 360.

Figure 18b is a schematic block diagram of a bridge device and one 10Mb/s collision domain and one 100Mb/s collision domain according to the present invention. The design shown in Figure 18b uses a different bandwidth (10Mb/s) in the first collision domain 382 as compared to the bandwidth (100Mb/s) in the second collision domain 360. The first collision domain 382 is coupled to the second collision domain 360 through the bridge device 356. The first collision domain includes a repeater 384, which is coupled to bridge device

356, DTE 386, DTE 388, DTE 390, and DTE 392. The second collision domain includes repeater 372, which is coupled to bridge device 356, DTE 374, DTE 376, DTE 378, and DTE 380.

Figure 19 is a schematic block diagram of a combined 100Base-T system according to the present invention. Such a system is built with repeaters and a bridge to deliver both 10Mb/s and 100Mb/s service. Bridge device 356 is coupled to SRAM 384 and SRAM 386. Bridge device 356 is coupled to TX transceiver 388 via MAC serial port 387. TX transceiver is coupled to 10 base repeater 390. 10 base repeater is coupled to PC 292 and PC 394. Bridge device 356 is coupled to repeater controller 398 via media independent interface (MII) port 396. Repeater controller 398 is coupled to TX transceiver 404 via TX port 400 and to TX transceiver 406 via TX port 402. Additional TX ports may be coupled to repeater controller 398 through ports 404. TX transceiver 404 is coupled to PC 412 via TX media 408. TX transceiver 406 is coupled to PC 414 via TX media 410.

Figure 20a is a schematic block diagram of a built-in bridge application according to the present invention. Bridge device 356 is coupled to SRAM 384 and SRAM 386. Bridge device 356 is coupled to TX clock recovery & transceiver 418 via TX port 416. TX clock recovery & transceiver 418 is coupled to TX media 420. Bridge device 356 is coupled to repeater controller 424 via MAC MII 422. Alternatively, a PHY MII / MAC MII interface can be used instead of MAC MII 422. Also, a PHY MII interface can be used instead of MAC MII 422.

Figure 20b is a schematic block diagram of a media independent interface (MII) repeater application according to the present invention. Bridge device 356 is coupled to repeater controller 428 via TX port 426. Repeater controller 428 is coupled to TX clock recovery & transceiver 434 via TX port 430. TX clock recovery & transceiver 434 is coupled to TX media 436. Bridge device 356 is coupled to TX clock recovery & transceiver 440 via TX port 438. TX clock recovery & transceiver 440 is coupled to TX media 442. Alternatively, instead of TX port 426, a PHY-MII port may be used to couple bridge device 356 to

repeater controller 428, and a MII port can be used to couple repeater controller 428 to TX clock recovery & transceiver 434.

Figure 20c is a schematic block diagram of a stand-alone bridge application according to the present invention. Bridge device 356 is coupled to SRAM 384 and SRAM 386. Bridge device 356 is coupled to TX clock recovery & transceiver 446 via TX port 444. TX clock recovery & transceiver 446 is coupled to TX media 448. Bridge device 356 is coupled to TX clock recovery & transceiver 452 via TX port 450. TX clock recovery & transceiver 452 is coupled to TX media 454. Alternatively, instead of TX clock recovery & transceiver 452, bridge device 356 may be coupled to a TX clock recovery chip, and the TX clock recovery chip would then be coupled to a FX fiber transceiver and the FX fiber transceiver would be coupled to an FX media. Also, instead of using TX port 450 and TX clock recovery & transceiver 452, bridge device 356 may be coupled to a 10/100-base twisted pair transceiver via an MII port and the 10/100-base twisted pair transceiver would then be coupled to a 10-base T media.

The following are tables showing pins in an integrated circuit embodiment of the invention:

Port A TX Symbol/Media Independent Interface

Pin Name	Type	Description
TDAT4_A	O	TX/FX Mode: Output. The most significant bit of the 5-bit Symbol Data.
TDAT3-0_A/RXD3-0_A	I/O	TX/FX Mode TDAT3-0_A: Output. These 4 pins and TDAT4_A are the 5-bit Symbol Data synchronous with TXCLK_A. MAC-MII Mode RXD3-0_A: Output. Synchronous with TXCLK_A, the 4-bit framed nibbles are used to transmit data to an external PHY device. PHY-MII Mode RXD3-0_A: Input. Synchronous with TXCLK_A, the 4-bit framed nibbles are the data transferred by an external MAC-like device.
TXCLK_A	I	TX/FX Mode: Input. It is a 25 MHz local symbol clock input. MII Mode: Input. It is a 2.5/25 MHz MII transmit clock input.
RDAT4_A	I	TX/FX Mode: Input. The most significant bit of the 5-bit Symbol Data.
RDAT3-0_A/RXD3-0_A	I/O	TX/FX Mode RDAT3-0_A: Input. These 4 pins and RDAT4_A are the 5-bit Symbol Data synchronous with RXCLK_A. MAC-MII Mode RXD3-0_A: Input. Synchronous with RXCLK_A, the 4-bit framed nibbles are used to receive data from an external PHY device. PHY-MII Mode RXD3-0_A: Output. Synchronous with RXCLK_A, the 4-bit framed nibbles are used to transfer data to an external MAC-like device. These pins are tri-stated if OE_A is asserted high.
RXCLK_A	I	TX/FX Mode: Input. The 25 MHz Receive Clock is generated by an external PHY device. MAC-MII Mode: Input. The Receive Clock generated by an external PHY device is 2.5/25 MHz in frequency.
SIGDET_A/COL_A	I/O	TX/FX Mode: Input. The Signal Detect indicates 10Base-T activity on the receive channel when asserted by an external PHY device. MAC-MII Mode: Input. The Collision signal from an external PHY device indicates that a receive carrier is detected while transmitting. PHY-MII Mode: Output. The Collision signal interfaces to an external MAC-like device, it is asserted if both the transmit and receive lines are active in half-duplex mode. This signal has no effect in full-duplex mode.
CRS_A	I/O	MAC-MII Mode: Input. The Carrier Sense signal is sourced by an external PHY device to indicate that either transmit or receive lines are active. PHY-MII Mode: Output. The Carrier Sense signal interfaces to an external MAC-like device. Note: In this mode, CRS_A and RXDV_A share the same driver inside the integrated circuit.
OE_A	I	Output Enable. Active low. It is used to control RXD[3:0], RXDV, RXER, RXCFHY_A in PHY_MII mode for use as output for port A.

Pin Name	Type	Description
RXCPHY_A	O	PHY-MII Mode: Output. It is a 2.5/25 MHz clock source. This pin is tri-stated if OE_A is asserted high.
RXDV_A	I/O	MAC-MII Mode: Input. The Receive Data Valid signal is generated by an external PHY device; it indicates that the recovered and decoded data nibbles are on the receive data lines. PHY-MII Mode: Output. The Receive Data Valid signal indicates to the external MAC-like device that integrated circuit is sending valid data nibbles for transmission. This pin can be tri-stated by pulling OE_A high. Note: In this mode, CRS_A and RXDV_A share the same driver inside the integrated circuit.
RXER_A	I/O	MAC-MII Mode: Input. The Receive Error signal is generated by an external PHY device. PHY-MII Mode: Output. The Receive Error signal connects to an external MAC-like device. This pin can be tri-stated by pulling OE_A high. Note: In this mode, the integrated circuit always drives a logic zero which indicates no receive error generated from the integrated circuit.
TXEN_A	I/O	MAC-MII Mode: Output. The Transmit Enable indicates that the integrated circuit is sending valid data nibbles for transmission to an external PHY device. PHY-MII Mode: Input. The Transmit Enable signal denotes that the recovered and decoded data nibbles are available on data lines from an external MAC-like device.

## Port B TX Symbol/Mode Independent Interface

Pin Name	Type	Description
TDAT4_B	O	TX/FX Mode: Output. The most significant bit of the 5-bit Symbol Data.
TDAT3-0_B/TXD3-0_B	I/O	TX/FX Mode TDAT3-0_B: Output. These 4 pins and TDAT4_B are the 5-bit Symbol Data synchronous with TXCLK_B. MAC-MII Mode: TXD3-0_B: Output. Synchronous with TXCLK_B, the 4-bit framed nibbles are used to transmit data to an external PHY device. PHY-MII Mode TXD3-0_B: Input. Synchronous with TXCLK_B, the 4-bit framed nibbles are the data transferred by an external MAC-like device.
		MAC-Serial Mode TXD0_B: Output. The serial transmit data bit which interfaces to an external PHY device. PHY-Serial Mode TXD0_B: Input. The serial data bit which interfaces to an external MAC-like device.
TXCLK_B	I	TX/FX Mode: Input. It is a 25 MHz local symbol clock input. MII Mode: Input. It is a 2.5/25 MHz MII transmit clock input.
		MAC/PHY-MII Serial Mode: It is a 10 MHz clock for 10Base serial mode.

Pin Name	Type	Description
RDAT4_B	I	TX/FX Mode: Input. The most significant bit of the 5-bit Symbol Data.
RDAT3-0_B/RXD3-0_B	I/O	<p>TX/FX Mode RDAT3-0_B: Input. These 4 pins and RDAT4_B are the 5-bit Symbol Data synchronous with TXCLK_B. MAC-MII Mode RXD3-0_B: Input. Synchronous with RXCLK_B, the 4-bit framed nibbles are used to receive data from an external PHY device. PHY-MII Mode RXD3-0_B: Output. Synchronous with RXCLK_B, the 4-bit framed nibbles are used to transfer data to an external PHY device. These pins are tri-stated when OE_B is asserted high.</p> <p>MAC-Serial Mode RXD0_B: Input. The receive serial data bit interfaces to an external PHY device. PHY-Serial Mode RXD0_B: Output. The serial data bit interfaces to an external MAC device.</p>
RXCLK_B	I	<p>TX/FX Mode: Input. The 25 MHz Received Clock is generated by an external PHY device. MAC-MII Mode: Input. The Receive Clock generated by an external PHY device is 2.5/25 MHz in frequency.</p> <p>MAC-Serial Mode: Input. It is a 10 MHz Receive Clock sourced by an external PHY device.</p>
SIGDET_B/COL_B	I/O	<p>TX/FX Mode: Input. The Signal Detect indicates 100Base-T activity on the receive channel when asserted by an external PHY device. MAC-MII Mode: Input. The Collision signal from an external PHY device indicates that a receive carrier is detected while transmitting. PHY-MII Mode: Output. The Collision signal interfaces to an external MAC-like device, it is asserted if both the transmit and receive lines are active in half-duplex mode. This signal has no effect in full-duplex mode.</p> <p>MAC-Serial Mode: Input. The Collision signal is provided by an external PHY device. PHY-Serial Mode: Output. The Collision signal interfaces to an external MAC-like device.</p>
CRS_B	I/O	<p>MAC-MII Mode: Input. The Carrier Sense signal is sourced by an external PHY device to indicate that either transmit or receive lines are active. PHY-MII Mode: Output. The Carrier Sense signal interfaces to an external MAC-like device. Note: In this mode, CRS_B and RXDV_B share the same driver inside the integrated circuit.</p> <p>MAC-Serial Mode: Input. The Carrier Sense signal interfaces to an external PHY device. PHY-Serial Mode: Output. This Carrier Sense signal interfaces to an external MAC-like device. Note: In this mode, CRS_B and RXDV_B share the same driver inside the integrated circuit.</p>
OE_B	I	Output Enable. Active low. It is used to control RGD(3:0), RGDV, RXER, PHYRXC_B in PHY-MII mode for use as output for port B.



Pin Name	Type	Description
RXCPHY_B	O	PHY-MII Mode: Output. A 2.5/25 MHz clock source. This pin is tri-stated if OE_B is asserted high.
		PHY-Serial Mode: Output. When interfacing to a MAC-like device (i.e. MAC-MII repeater), the RXCPHY and TXCLK of both the integrated circuit and the MAC-MII repeater are sourced by the same external 10 MHz clock.
RXDV_B	I/O	MAC-MII Mode: Input. The Receive Data Valid signal is generated by an external PHY device; it indicates that the recovered and decoded data nibbles are on the receive data lines. PHY-MII Mode: Output. The Receive Data Valid signal indicates to the external MAC-like device that the integrated circuit is sending data nibbles for transmission. This output can be tri-stated by pulling OE_B high. Note: In this mode, CRS_B and RXDV_B share the same driver inside the integrated circuit.
		MAC-MII Mode: Input. The Receive Error signal is generated by an external PHY device. PHY-MII Mode: Output. The Receive Error signal connects to an external MAC device. This pin can be tri-stated by pulling OE_B high. Note: In this mode, the integrated circuit always drives a logic zero which indicates no receiver error generated from the integrated circuit.
TXEN_B	I/O	MAC-MII Mode: Output. The Transmit Enable indicates that the integrated circuit is sending valid data nibbles for transmission to an external PHY device. PHY-MII Mode: Input. The Transmit Enable signal denotes that the recovered and decoded data nibbles are available on data lines from an external MAC-like device.
		MAC-Serial Mode: Output. The Transmit Enable signal interfaces to an external PHY device. PHY-Serial Mode: Input. The Transmit Enable signal interfaces to an external MAC-like device.

## Mode Configuration Pins

Pin Name	Type	Description
MSIZE1-0	I	Memory Size. Mut'd with ABYTE1-0_B. [MSIZE1, MSIZE0] = Memory size [0, 0] = 16 Kbytes; [0, 1] = 64 Kbytes; [1, 0] = 128 Kbytes; [1, 1] = 256 Kbytes.
MRAT2-0	I	Memory Ratio. These two pins define the buffer size ratio of Port A to Port B. A larger buffer at Port A means Port A is capable of receiving more packets to be forwarded to Port B. [MRAT2, MRAT0] = Port A buffer size : Port B buffer size [x, x, 0] = 1:1; [0, 0, 1] = 1:7; [0, 1, 1] = 1:15; [1, 0, 1] = 7:1; [1, 1, 1] = 15:1.

Pin Name	Type	Description
AUTOSIZE	I	Auto Size. Mux'd with ABYTE1_A. When asserted high, it enables Auto-size feature of buffer memory.
FWD1-0_A	I	Port A Forward Mode Selection. [FWD1_A, FWD0_A] - Forward Mode [0, 0] - Half-duplex, full-packet store-and-forward; [0, 1] - Full-duplex, full-packet store-and-forward; [1, 0] - Half-duplex, 64-byte store-and-forward; [1, 1] - Full-duplex, 64-byte store-and-forward.
FWD1-0_B	I	Port B Forward Mode Selection. [FWD1_B, FWD0_B] - Forward Mode [0, 0] - Half-duplex, full-packet store-and-forward; [0, 1] - Full-duplex, full-packet store-and-forward; [1, 0] - Half-duplex, 64-byte store-and-forward; [1, 1] - Full-duplex, 64-byte store-and-forward.
SPEED_A	I	Port A Speed. When asserted high, 100 M speed is employed for Port A. When asserted low, 10 M speed is employed.
SPEED_B	I	Port B Speed. When asserted high, 100 M speed is employed for Port B. When asserted low, 10 M speed is employed.
PHY1-0_A	I	PHY of Port A. Mux'd with ABYTE1-0_A. [PHY1_A, PHY0_A] - [0, 0] - MAC-MII (interfacing to an external PHY device); [0, 1] - PHY-MII (interfacing to an external MAC device); [1, 0] - 100BASE-FX; [1, 1] - 100BASE-TX.
PHY2-0_B	I	PHY of Port B. [PHY2_B, PHY1_B, PHY0_B] - [0, 0, 0] - MAC-MII (interfacing to an external PHY device); [0, 0, 1] - PHY-MII (interfacing to an external MAC device); [0, 1, 0] - 100BASE-FX; [0, 1, 1] - 100BASE-TX; [1, 0, 0] - MAC-Serial (interfacing to an external PHY device); [1, 0, 1] - PHY-Serial (interfacing to an external MAC device); [1, 1, 0] - Reserved; [1, 1, 1] - Reserved.
FCEN_A	I	Flow Control Enable. Mux'd with Buf1d_A. When asserted high, it enables flow control mechanism in full-duplex mode on Port A.
FCEN_B	I	Flow Control Enable. Mux'd with Buf1d_B. When asserted high, it enables flow control mechanism in full-duplex mode on Port B.

## Address Filter Pin

Pin Name	Type	Description
SLNP_AB	I	Mux'd with PKDT7_B
BCF_AB	I	Mux'd with PKDT6_B
MCF_AB	I	Mux'd with PKDT5_B
SAF_AB	I	Mux'd with PKDT4_B
SLNP_BA	I	Mux'd with PKDT3_B
BCF_BA	I	Mux'd with PKDT2_B
MCF_BA	I	Mux'd with PKDT1_B

Pin Name	Type	Description
SAF_BA	I	Mux'd with PKDT0_B.

**Buffer SRAM Interface**

Pin Name	Type	Description
SA16-0	O	SRAM Address 16-0. These 17 address inputs select one of the 128K 16-bit words in the SRAM.
SD15-0	I/O	SRAM Data Input/Output Ports. These 16 bidirectional pins are used to read data from or write data into the SRAM.
SR_W	O	Read Not Write. When asserted high, it enables the read operation of SRAMs. When asserted low, it enables the write operation.
SO_	O	Output Enable. It is an active low signal which controls outputs from SRAMs.

**Register Interface**

Pin Name	Type	Description
REGDAT[7:0]	I/O	Register Data. These are the 8-bit register data. They are tri-stated if CS_ is not asserted.
REGSEL[7:0]	I	Register Select. These pins are used to select internal registers.
CS_	I	Chip Select. When asserted low, it enables the read or write access to the register.
REGR_W	I	Register Read Not Write. When asserted high, it enables read operation. When asserted low, it enables write operation.

**EEPROM Interface Pins**

Pin Name	Type	Description
EECS/NOEEPROM	I/O	EEPROM Chip Select. Output. It is asserted high during a EEPROM read cycle. NOEEPROM. Input. It should be pulled high when there is no EEPROM.
EESK	O	EEPROM Clock.
EEDI	O	EEPROM Serial Data Input. It is connected to the serial data in of the EEPROM.
EEDO	I	EEPROM Serial Data Output. It is connected to the serial data out of the EEPROM.

Address Filed Access Pins

Pin Name	Type	Description
PKTD7-0_A/AGTM7-0	I/O	<p>Packet Data From Port A. Output. PKTD7 is the MSB in a byte. The order displayed is as follows: Idle-preamble-SFD (1 byte) - DA (6 bytes) - Hashed DA (1 byte) - SA (6 bytes) - Hashed SA (1 byte) - Data - Idle.</p> <p>AGTM7-0. Input. The value of aging timer is read into the integrated circuit during power-up reset. AGTM7 is the MSB and ATTM0 is the LSB. The aging timer value is minimum 3 minutes and maximum 1275 minutes.</p>
ABYTE2_A/AUTO-SIZE ABYTE1_A/PHY1_A ABYTE0_A/PHY0_A	I/O,TTL	<p>Address Byte From Port A. Output. These three pins indicate the content of PKTD7-0_A. [ABYTE2, ABYTE1, ABYTE0] [0, 0, 0] = PKTD7-0_A is the SFD (10101011); [0, 0, 1] = PKTD7-0_A are the DA bytes; [0, 1, 0] = PKTD7-0_A are the Most Significant 8 bits of the CRC of the 6 DA bytes; [0, 1, 1] = PKTD7-0_A are the SA bytes; [1, 0, 0] = PKTD7-0_A are the Most Significant 8 bits of the CRC of the 6 SA bytes; [1, 0, 1] = PKTD7-0_A are the data; [1, 1, 0] = PKTD7-0_A are the Idle state; [1, 1, 1] = PKTD7-0_A are the preamble. AutoSize. Input. When asserted high, it enables the auto-size feature of buffer memory. PHY1-0_A. Input. Physical interface for Port A.</p>
ABORT_A	I	<p>Abort Packet From Port A. When this pin is asserted high, it signals the integrated circuit to abort a most recently received packet. If this packet is stored in the buffer, the buffer is reclaimed.</p>
PKTD7-0_B	I/O	<p>Packet Data From Port B. Output. PKTD7 is the MSB in a byte. The order displayed is as follows: Idle-preamble-SFD (1 byte) - DA (6 bytes) - Hashed DA (1 byte) - SA (6 bytes) - Hashed SA (1 byte) - Data - Idle.</p>
SLENP_AB		Input. Active high signal. Self Learning Algorithm from A to B Enabled.
BCF_AB		Broadcast Filter from A to B Enabled.
MCF_AB		Multicast Filter from A to B Enabled.
SAP_AB		Self-Addressed Packet Filter from A to B Enabled.
SLENP_BA		Self Learning Algorithm from B to A Enabled.
BCF_BA		Broadcast Filter from B to A Enabled.
MCF_BA		Multicast Filter from B to A Enabled.
SAP_BA		Self-Addressed Packet Filter from B to A Enabled.

Pin Name	Type	Description
ABYTE2_B ABYTE1_B/MSIZE1 ABYTE0_B/MSIZE0	O I/O	Address Byte From Port B. Output. These three pins indicate the content of PKTD7-0_B. (ABYTE2, ABYTE1, ABYTE0) {0, 0, 0} = PKTD7-0_B is SFD (10101011); {0, 0, 1} = PKTD7-0_B are DA bytes; {0, 1, 0} = PKTD7-0_B are the Most Significant 8 bits of the CRC of the 6 DA bytes; {0, 1, 1} = PKTD7-0_B are the SA bytes; {1, 0, 0} = PKTD7-0_B are the Most Significant 8 bits of the CRC of the 6 SA bytes; {1, 0, 1} = PKTD7-0_B are the data; {1, 1, 0} = PKTD7-0_B are the Idle state; {1, 1, 1} = PKTD7-0_B are the preamble.  MSIZE1-0. Input. These two pins select buffer memory size.
ABORT_B	I	Abort Packet From Port B. When ABORT_B is asserted high, it signals the integrated circuit to abort the most recently received packet. If this packet is stored in the buffer, the buffer is reclaimed.

## Buffer Full Indication

Pin Name	Type	Description
BUFUL_A/FCEN_A	I/O	Buffer Full At Port A. Output. When BUFUL_A is asserted high, it indicates that buffer is full due to the received packets on Port A. If desired, it may be used with 74LS244 to drive an LED as the Port A buffer status indication. Flow Control Enable. Input. Flow control mechanism enable pin.
BUFUL_B/FCEN_B	I/O	Buffer Full At Port B. Output. When BUFUL_B is asserted high, it indicates that buffer is full due to the received packets on Port B. If desired, it may be used with 74LS244 to drive an LED as the Port B buffer status indication. Flow Control Enable. Input. Flow control mechanism enable pin.

In sum, the present invention provides a device that helps to reduce the complexity and costs associated with a bridge and helps to more efficiently provide the functionality of interconnection between network media and helps to avoid leaks of local packets to other network segments and is scalable for higher speed networks.

The foregoing description of a preferred embodiment of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obviously, many modifications and variations will be apparent to practitioners skilled in this

art. It is intended that the scope of the invention be defined by the following claims and their equivalents.

## CLAIMS

What is claimed is:

1. A device for coupling a first network medium to a second network medium, the device comprising:
  - a first port coupled to the first network medium;
  - a second port coupled to the second network medium;
  - a memory storing a first plurality of indications and a second plurality of indications, the indications in the first plurality of indications corresponding to respective sets of addresses and indicating whether at least one address in the respective set of addresses may be accessible through the first network medium, the indications in the second plurality of indications corresponding to respective sets of addresses and indicating whether at least one address in the respective set of addresses may be accessible through the second network medium; and
  - a connecting circuit coupled to the first port, the second port, and the memory, the connecting circuit causing the ports to pass or block a packet from the first network to the second network, the packet having a destination address, the connecting circuit causing the ports to pass or block based on:
    - a first indication from the first plurality of indications, the first indication corresponding to a set of addresses including the destination address of the packet, and
    - a second indication from the second plurality of indications, the second indication corresponding to the set of addresses including the destination address of the packet.
2. The device of claim 1, further comprising:
  - a first circuit that reads a source address of the packet, the first circuit setting a third indication in the first plurality of indications, the third indication corresponding to the source address of the packet.
3. The device of claim 2, further comprising:

a second circuit that reads a source address of a second packet from the second port, the second circuit setting a fourth indication in the second plurality of indications, the fourth indication corresponding to the source address of the second packet.

4. The device of claim 2, further comprising  
a central processing unit (CPU) interface coupled to the memory that allows a CPU to set indications in the memory.
5. The device of claim 2, further comprising:  
an erase circuit coupled to the memory that erases old entries from the memory.
6. The device of claim 2, further comprising:  
an aging timer;  
an erase circuit coupled to the aging timer and the memory;  
and wherein the first plurality of indications comprises:  
a third plurality of indications, and  
a fourth plurality of indications;  
and wherein the first circuit is configured to set the third indication in the third plurality of indications or in the fourth plurality of indications based on the aging timer;  
and wherein the erase circuit is configured to erase the third plurality of indications or the fourth plurality of indications based on the aging timer.
7. The device of claim 3, further comprising:  
an aging timer;  
an erase circuit coupled to the aging timer and the memory;  
and wherein the first plurality of indications comprises:  
a third plurality of indications, and  
a fourth plurality of indications;



and wherein the second plurality of indications comprises:

a fifth plurality of indications, and

a sixth plurality of indications;

and wherein the first circuit is configured to set the third indication in the third plurality of indications or in the fourth plurality of indications based on the aging timer;

and wherein the second circuit is configured to set the fourth indication in the fifth plurality of indications or in the sixth plurality of indications based on the aging timer;

and wherein the erase circuit is configured to erase the third plurality of indications or the fourth plurality of indications based on the aging timer and the erase circuit is configured to erase the fifth plurality of indications or the sixth plurality of indications based on the aging timer.

8. The device of claim 1, wherein the connecting circuit is configured to block the packet from the first network segment to the second network segment if:

the first indication indicates that at least one address in the second set of addresses may be accessible through the first network medium; and

the second indication does not indicate that at least one address in the fourth set of addresses may be accessible through the second network medium.

9. The device of claim 1, wherein the connecting circuit is configured to pass the packet from the first network segment to the second network segment if the second indication indicates that the destination address of the packet may be accessible through the second network segment.

10. The device of claim 1, further comprising:

an index generation circuit that generates a first index in response to the destination address of the packet; and

a selector circuit that selects the first indication from the first plurality of indications based on the first index, the selector circuit coupled to the connecting circuit and to the index generation circuit.

11. The device of claim 10, wherein the index generation circuit comprises:  
a hash circuit and wherein the first index comprises a result of a hash function of the destination address of the packet.

12. The device of claim 11, wherein the hash circuit comprises:  
a cyclic redundancy check (CRC) circuit.

13. The device of claim 1, wherein the respective sets of addresses comprise medium access control (MAC) addresses.

14. The device of claim 1, wherein the first network medium comprises a network using carrier sense multiple access protocol.

15. The device of claim 1, wherein the first network medium comprises a collision domain network segment.

16. The device of claim 1, wherein the entire device is implemented on a single integrated circuit.

17. A method of selectively passing packets from a first network medium to a second network medium, the method comprising the steps of:

passing or blocking a first packet from the first network segment to the second network segment based on a first indication and a second indication;  
wherein the first indication corresponds to a destination address of the first packet and indicates whether an address in a first set of address including the destination address may be accessible through the first network medium; and

wherein the second indication corresponds to the destination address of the first packet and indicates whether an address in a second set of addresses including the destination address may be accessible through the second network medium.

18. The method of claim 17, wherein the step of passing or blocking a first packet from the first network segment based on a first indication and a second indication further comprises:

passing the first packet from the first network segment to the second network segment if the second indication indicates that an address in the second set of addresses may be accessible through the second network medium.

19. The method of claim 17, wherein the step of passing or blocking a first packet from the first network segment based on a first indication and a second indication further comprises:

blocking the first packet from the first network segment to the second network segment if:

the first indication indicates that an address in the first set of addresses may be accessible through the first network medium; and

blocking the first packet from the first network segment to the second network segment if the second indication does not indicate that the address in the second set of addresses may be accessible through the second network medium.

20. The method of claim 17, further comprising the steps of:

performing a hash function on the address of the node in the first network segment to yield a first hash value; and

indexing the first indication by the first hash value.

21. The method of claim 20, wherein the step of performing a hash function comprises performing a cyclic redundancy check on the address of the node in the first network segment.

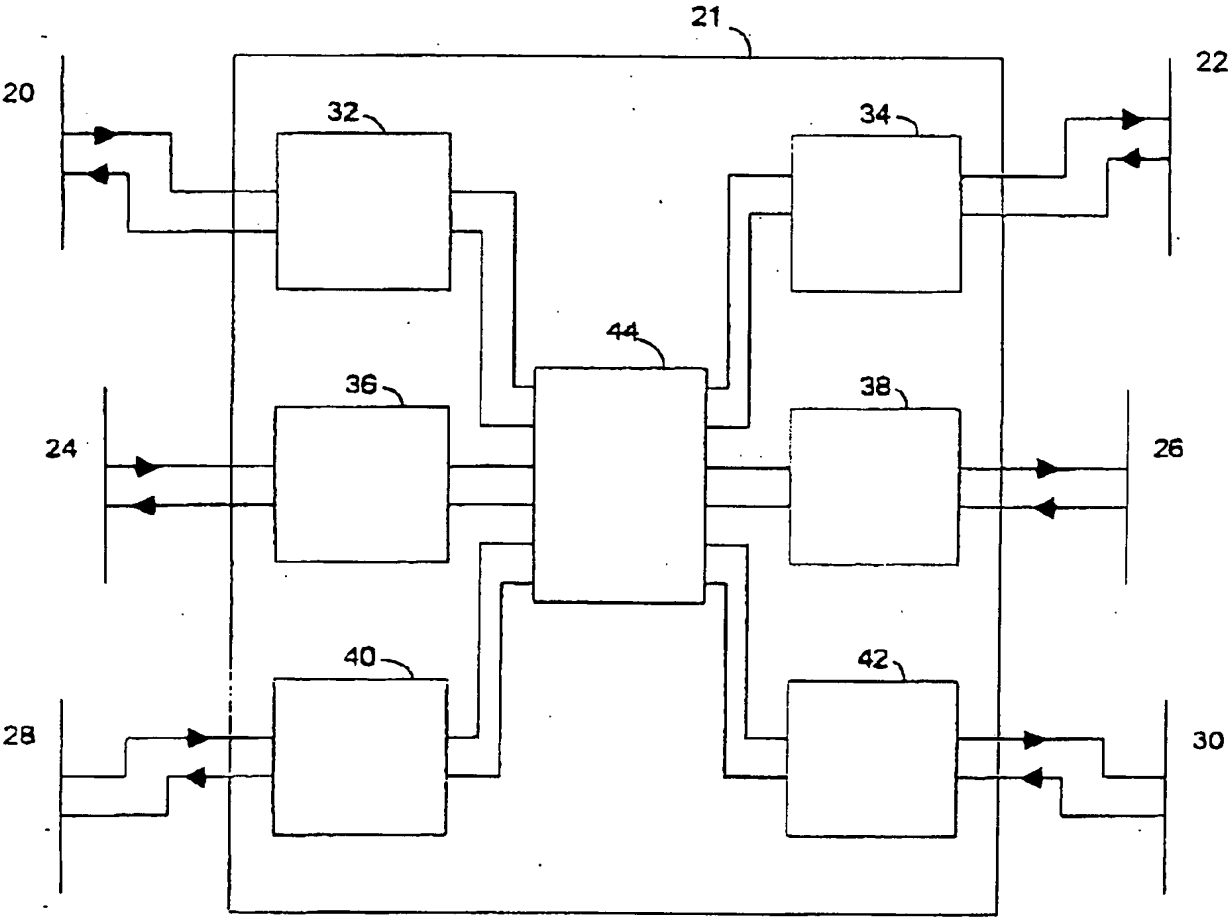


FIG.-1

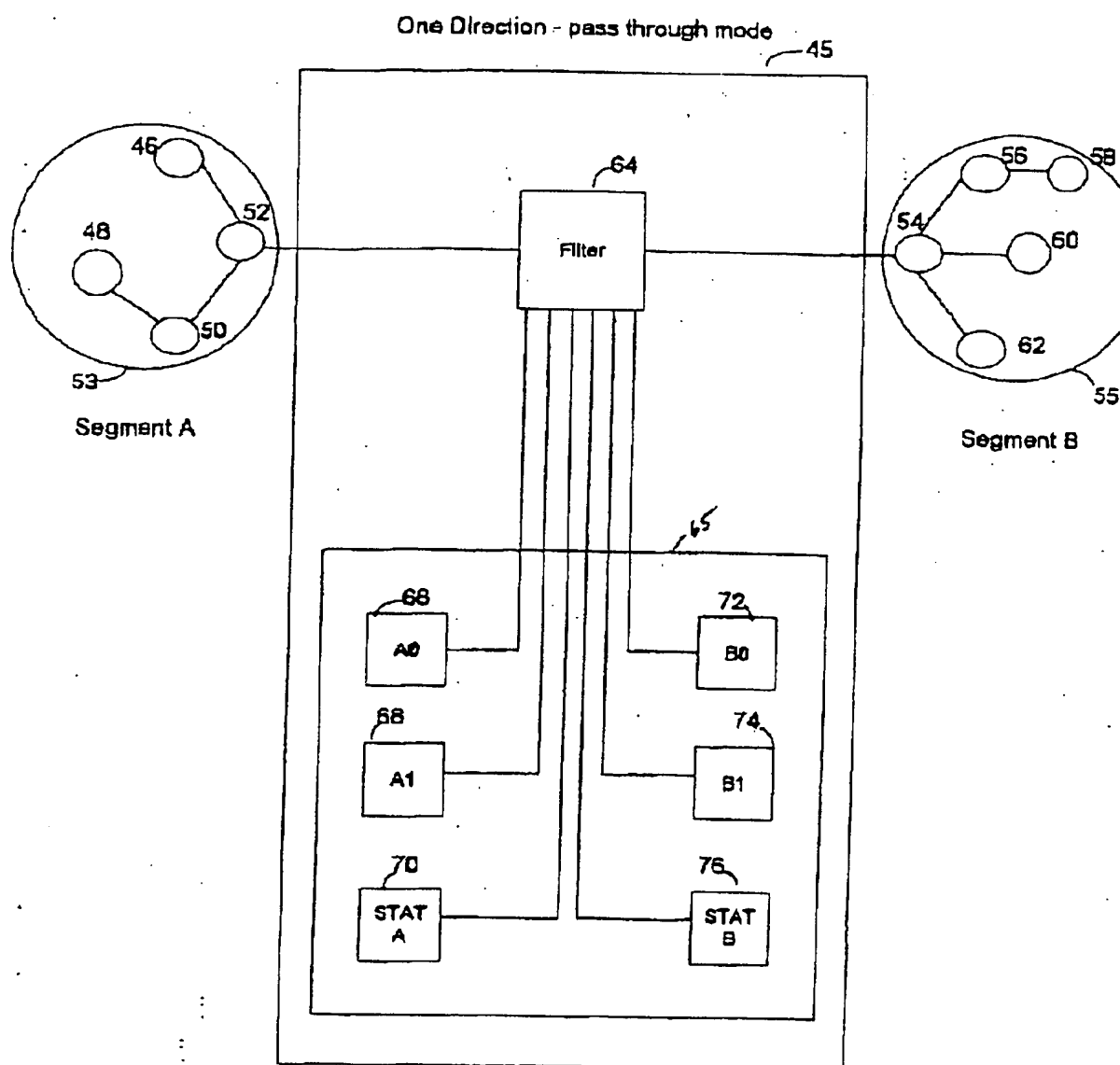


FIG.-2

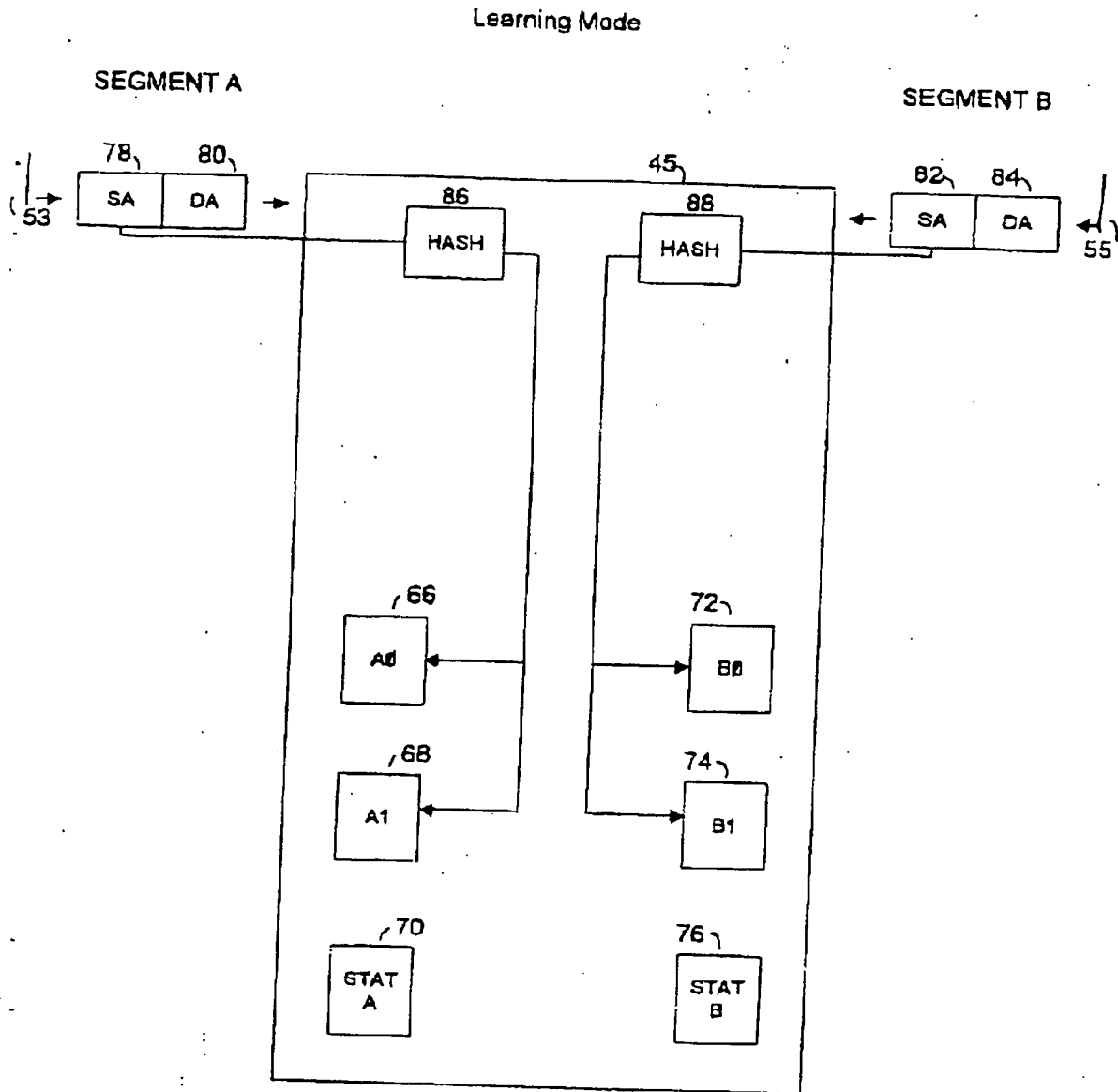


FIG.-3

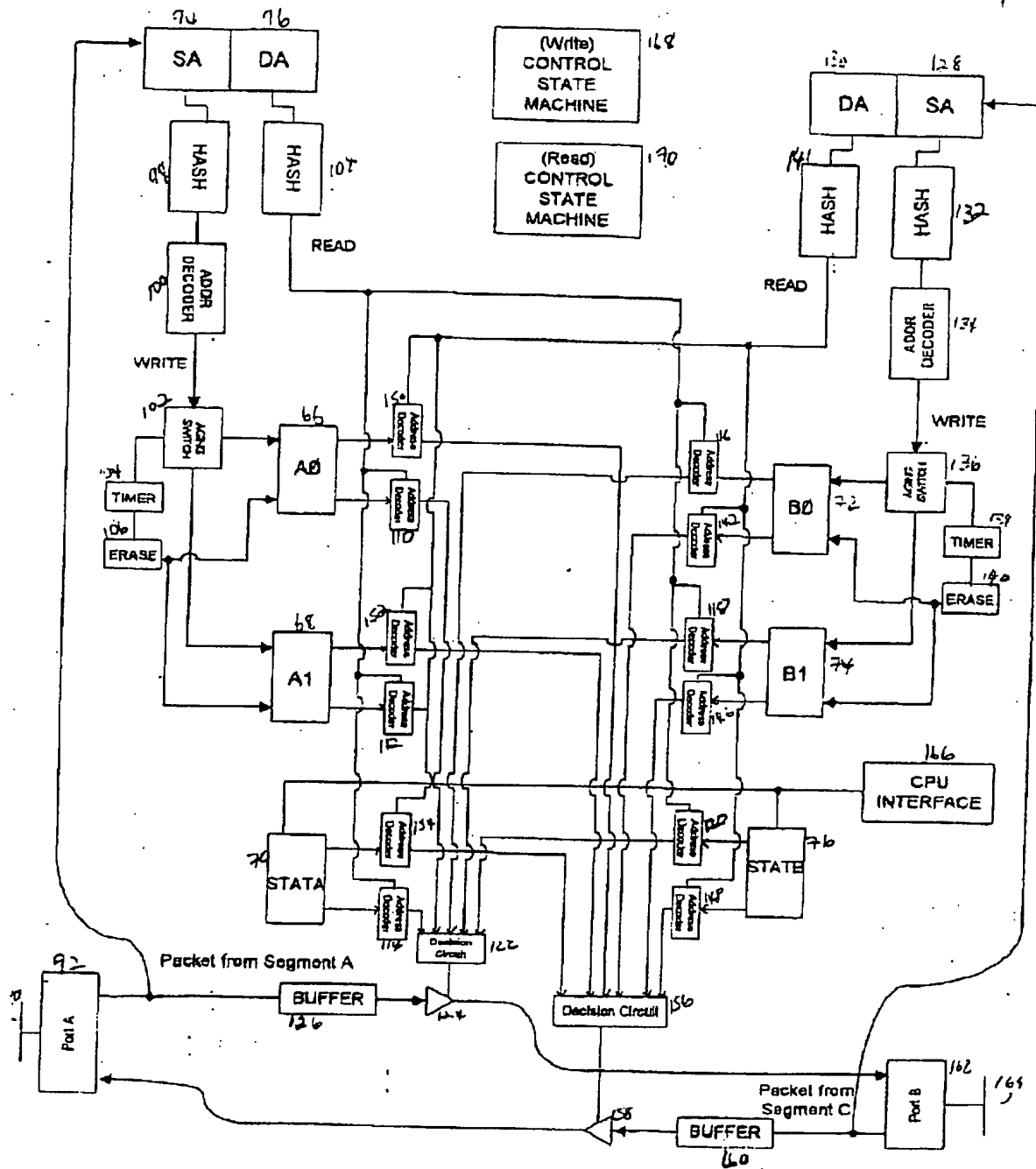


FIG. - 4



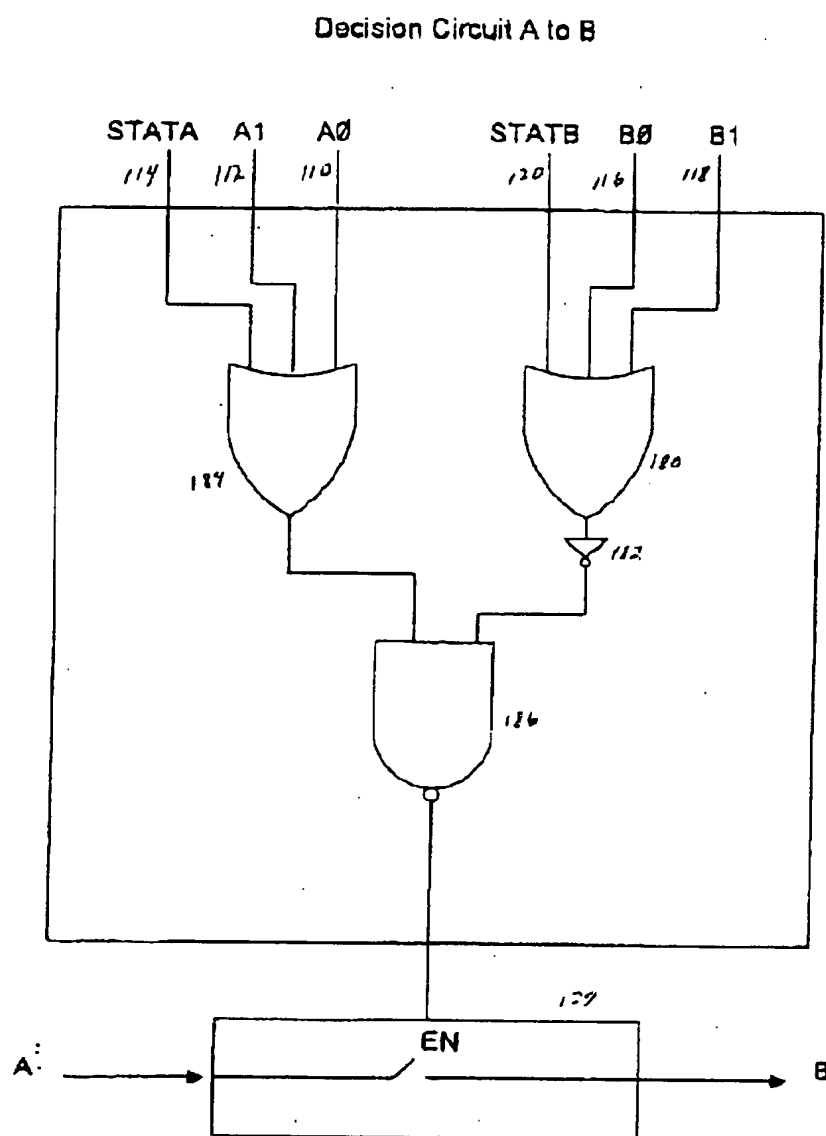
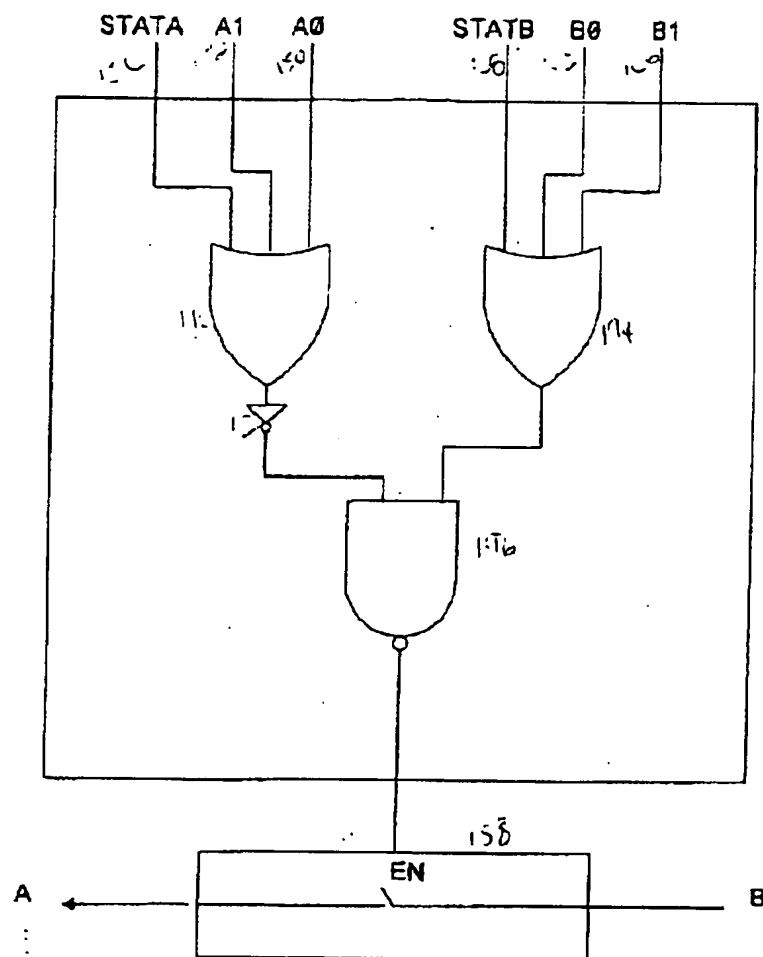


FIG.-5

**FIG.-6**

X = 0 or 1

A → B

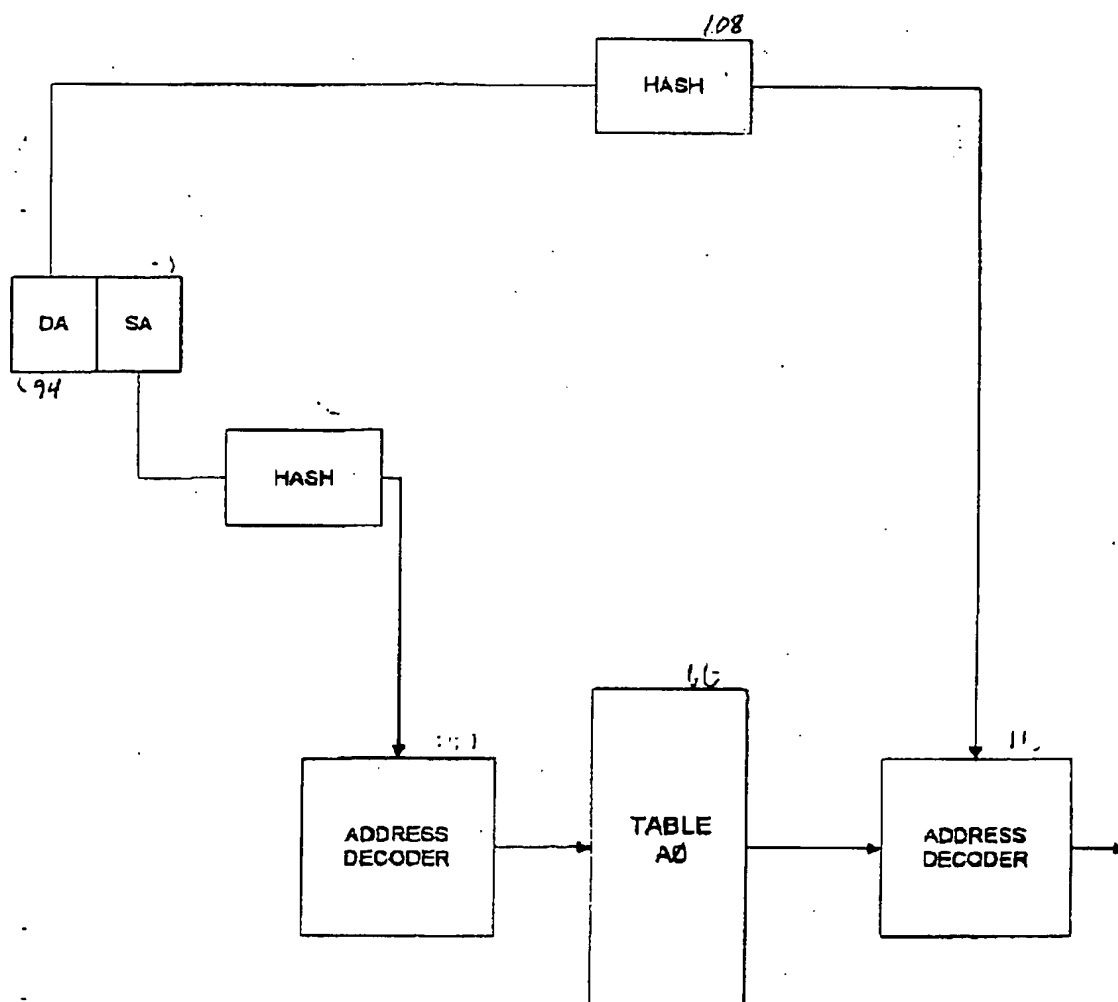
A0	A1	STATA	B0	B1	STATB	EN (1 = Pass to B)
0	0	0	X	X	X	1
1	X	X	0	0	0	0
X	1	X	0	0	0	0
X	X	1	0	0	0	0
X	X	X	1	X	X	1
X	X	X	X	1	X	1
X	X	X	X	X	1	1

B → A

B0	B1	STATB	A0	A1	STATA	EN (1 = Pass to A)
0	0	0	X	X	X	1
1	X	X	0	0	0	0
X	1	X	0	0	0	0
X	X	1	0	0	0	0
X	X	X	1	X	X	1
X	X	X	X	1	X	1
X	X	X	X	X	1	1

FIG.-7

**FIG.-8**

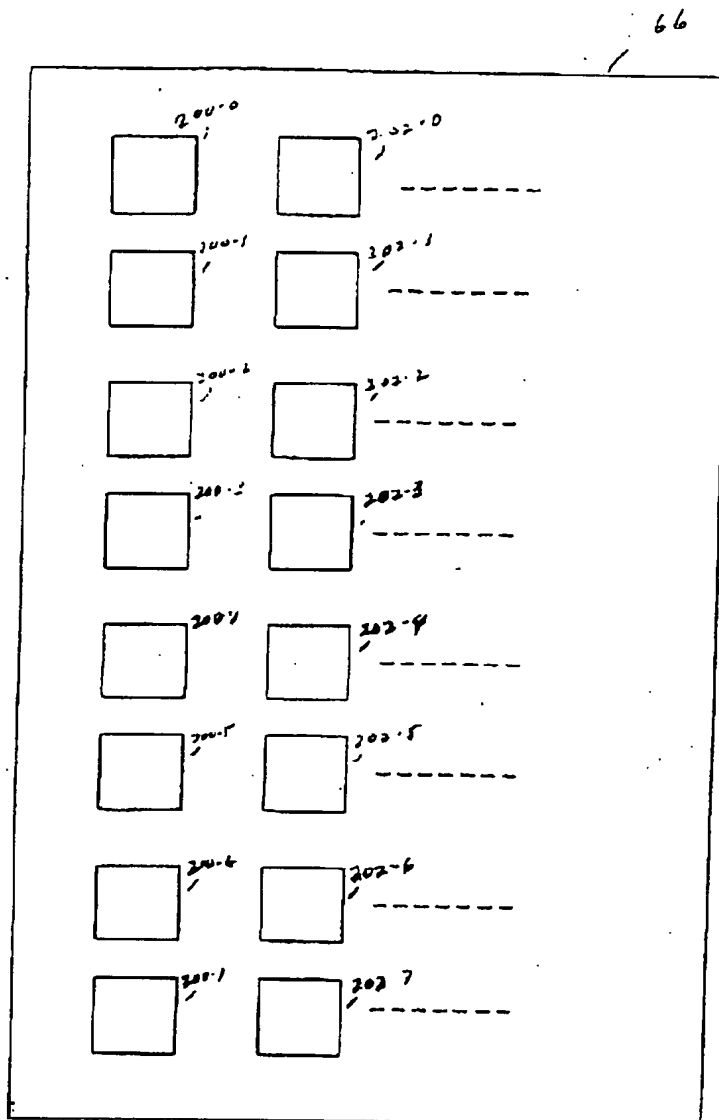
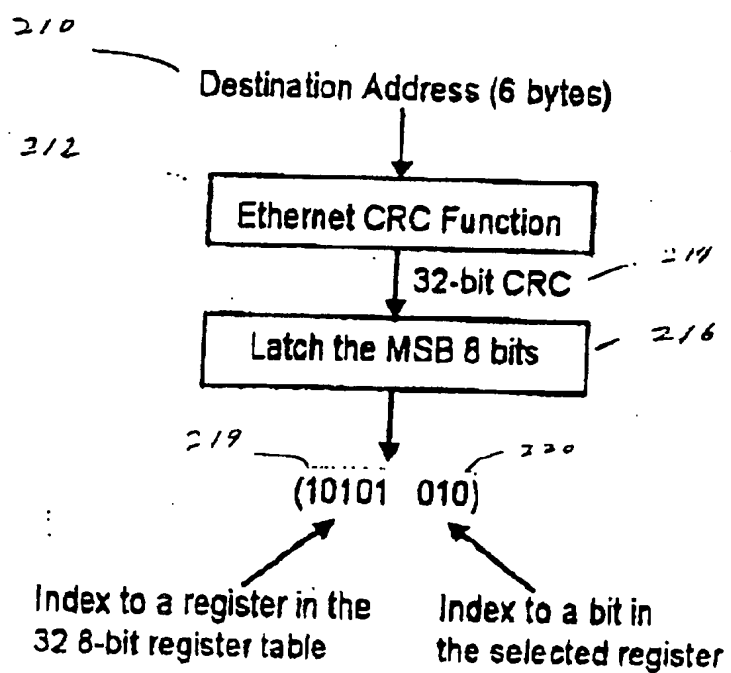
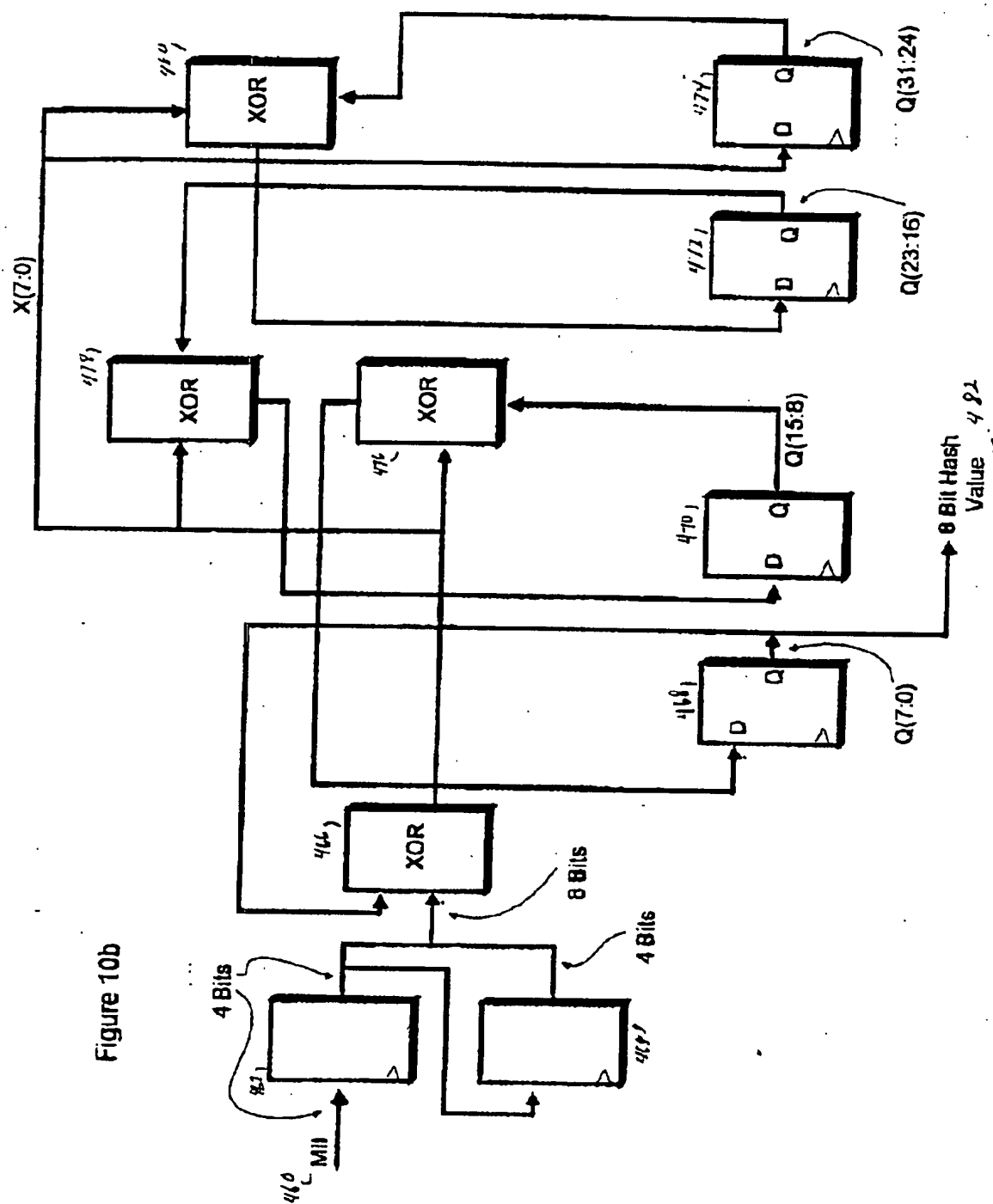


FIG.-9

**FIG. 10a**



	B	M	S	IB	IM	H	Packets to Be Blocked
1	0	0	0	0	0	0	None
2	0	0	0	0	0	1	Static hash-filtered packets EXCEPT broadcast/multicast packets
3	1	0	0	0	0	0	Broadcast packets
4	1	0	0	0	0	1	Broadcast AND static hash-filtered packets
5	0	1	0	0	0	0	Multicast (but not broadcast) packets
6	0	1	0	0	0	1	Multicast (but not broadcast) AND static hash-filtered packets
7	0	0	1	0	0	0	Self-addressing packets
8	0	0	1	0	0	1	Self-addressing AND static hash-filtered packets
9	1	1	0	0	0	0	Broadcast AND multicast packets
10	1	1	0	0	0	1	Broadcast, multicast AND static hash-filtered packets
11	1	0	1	0	0	0	Broadcast AND self-addressing packets
12	1	0	1	0	0	1	Broadcast, self-addressing AND static hash-filtered packets
13	0	1	1	0	0	0	Multicast (but not broadcast), self-addressing packets
14	0	1	1	0	0	1	Multicast (but not broadcast), self-addressing & static hash-filtered packets
15	1	1	1	0	0	0	Broadcast, multicast, AND self-addressing packets
16	1	1	1	0	0	1	Broadcast, multicast, self-addressing AND static hash-filtered packets
17	x	x	x	1	0	x	Packets with DA != FFFFFFFF
18	x	x	x	0	1	x	Broadcast AND packets with the first bit of DA != 1
19	x	x	x	1	1	x	Packets with the first bit of DA != 1

Note: B: Broadcast Packet Filter, M: Multicast Packet Filter, S: Self-Addressing Packet Filter,  
H: 256-bit Hash Filter, IB: Inverse Broadcast Filter, IM: Inverse Multicast Packet

FIG. 11



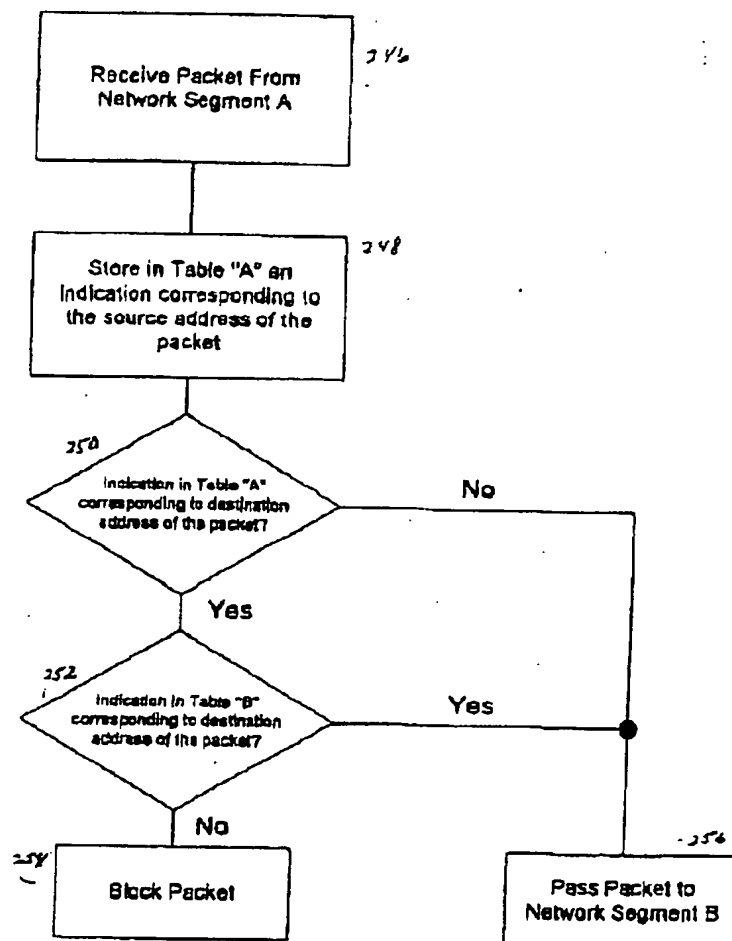


FIG.-12

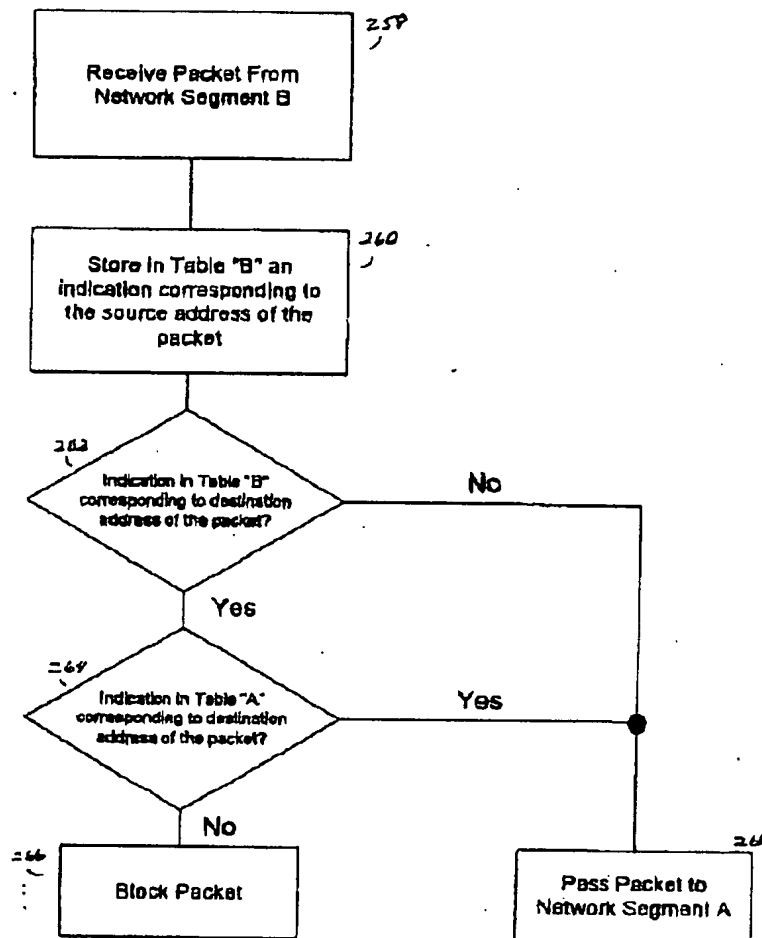


FIG.-13

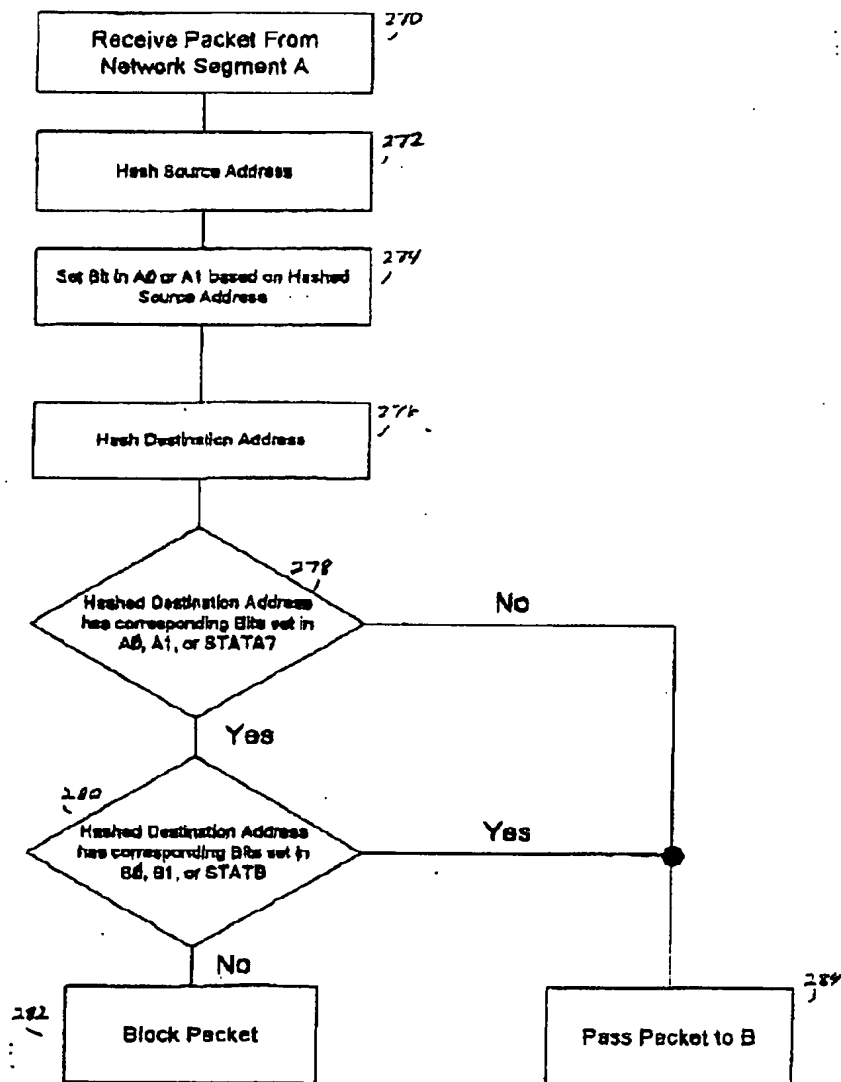


FIG.-14

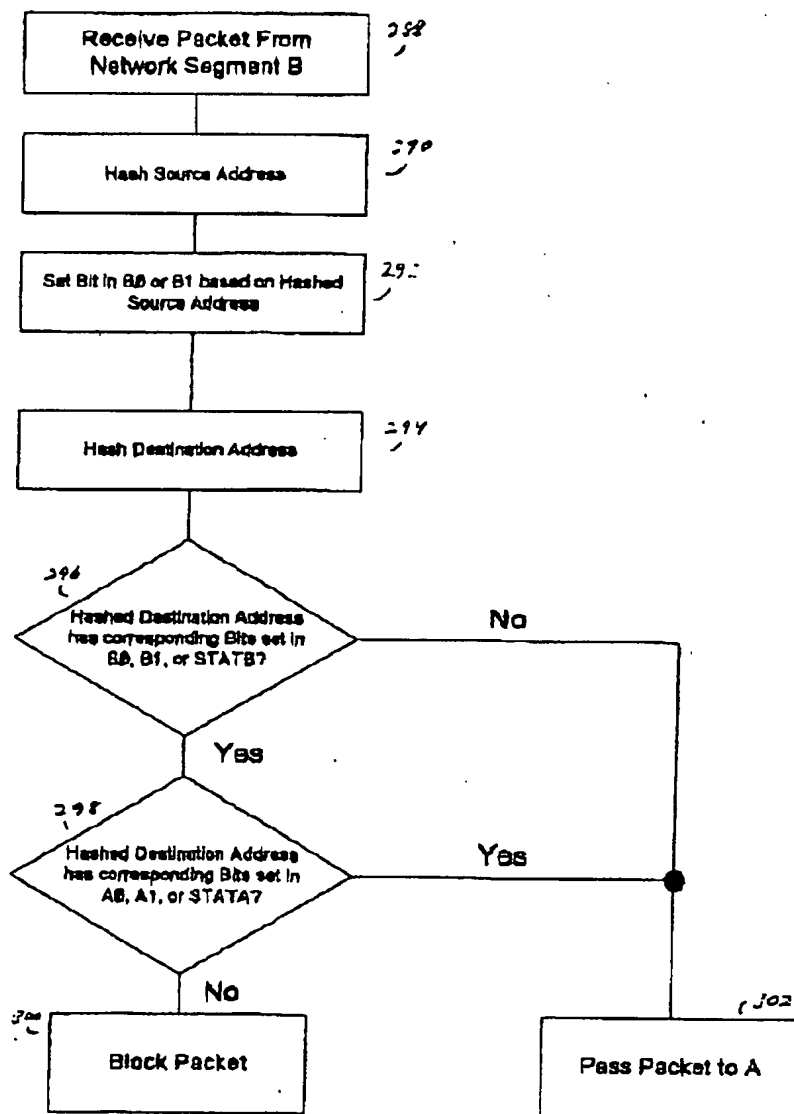


FIG.-15

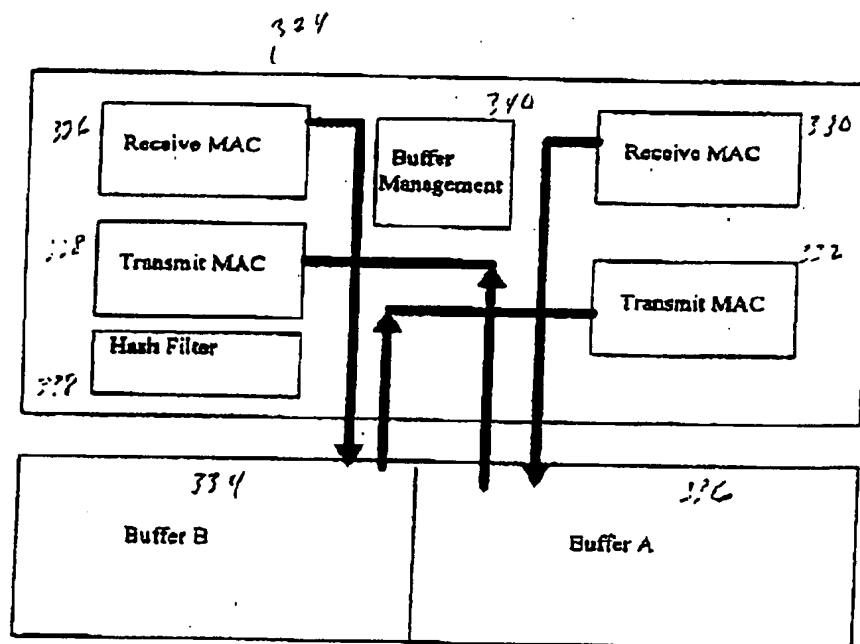


FIG. 16

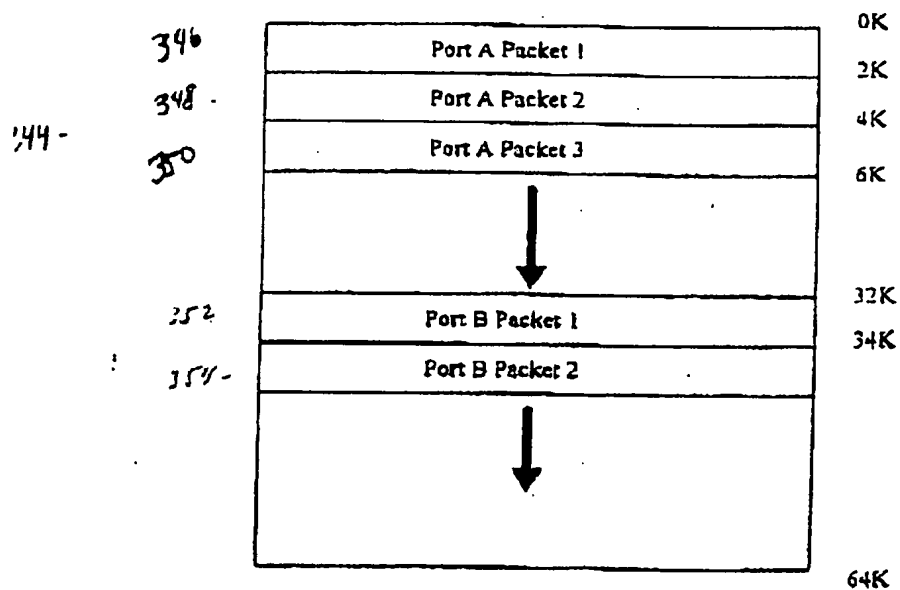


FIG. 17

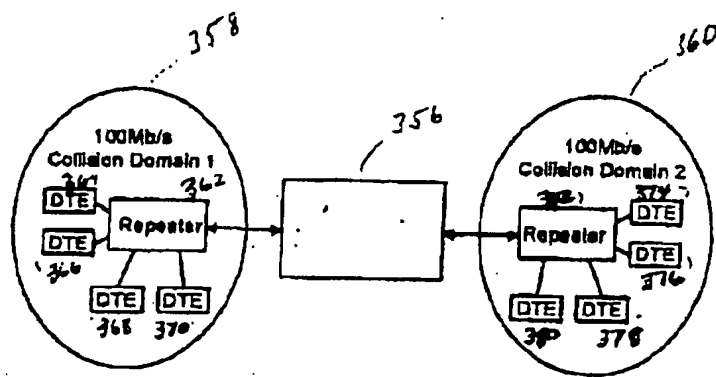


FIG. 18a

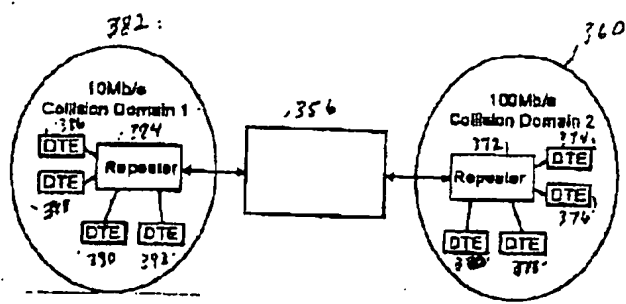


FIG. 18b



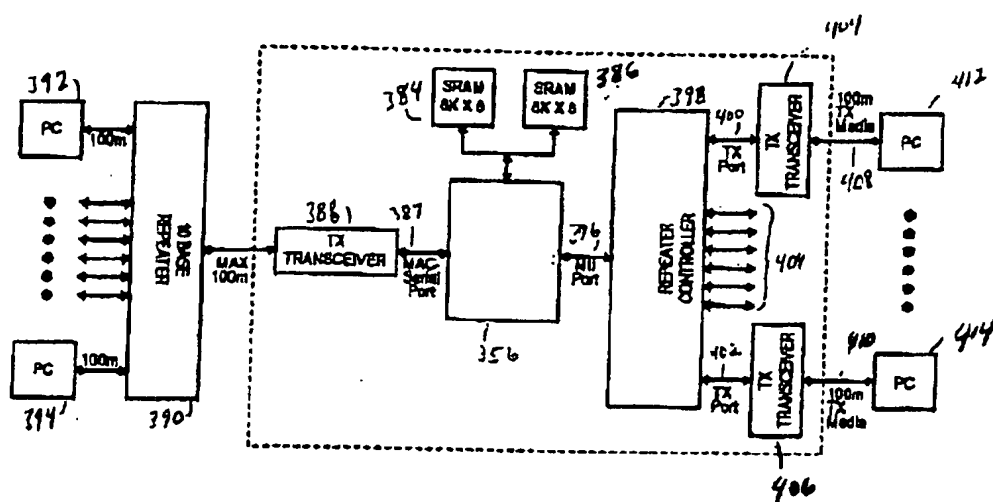
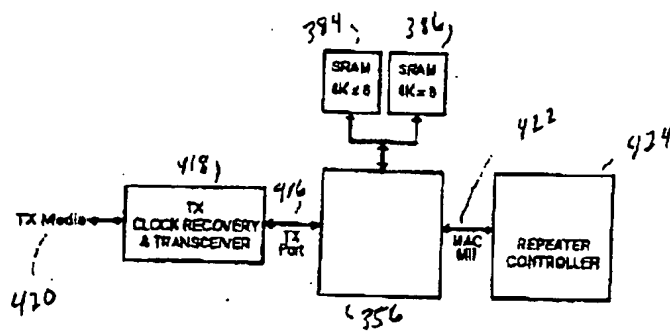
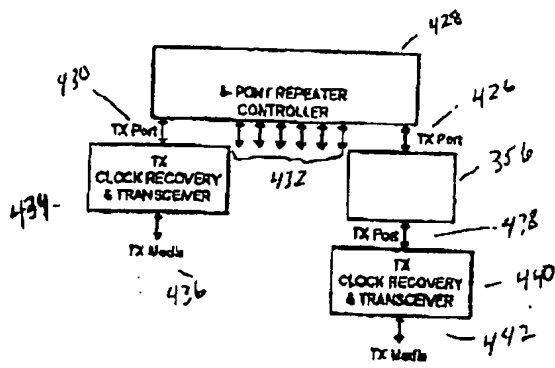
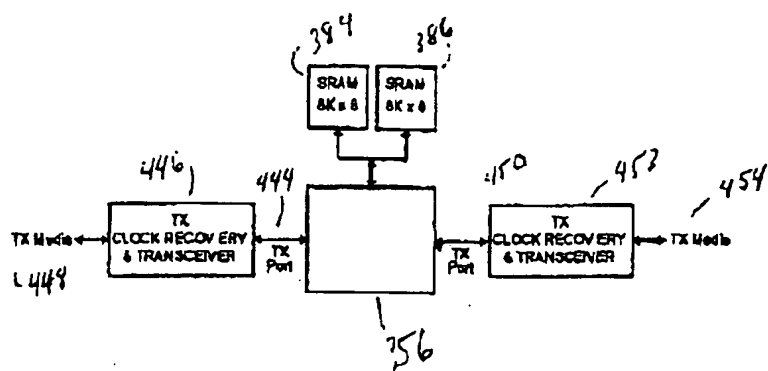


FIG. 19

**FIG. 20a**



**FIG. 20b**

**FIG. 20c**

**BRIDGE DEVICE WITH SELF LEARNING BETWEEN NETWORK  
MEDIA AND INTEGRATED CIRCUIT AND METHOD BASED ON  
THE SAME**

**ABSTRACT**

A device for coupling a first network medium to a second network medium. A first port is coupled to the first network medium and a second port coupled to the second network medium. A memory stores a first plurality of indications and a second plurality of indications. The indications in the first plurality of indications correspond to respective sets of addresses and indicate whether at least one address in the respective set of addresses may be accessible through the first network medium. The indications in the second plurality of indications correspond to respective sets of addresses and indicate whether at least one address in the respective set of addresses may accessible through the second network medium. A connecting circuit is coupled to the first port, the second port, and the memory. The connecting circuit causes the ports to pass or block a packet from the first network to the second network. The packet has a destination address. The connecting circuit causes the ports to pass or block based on a first indication from the first plurality of indications and a second indication from the second plurality of indications. The first indication corresponds to a set of addresses including the destination address of the packet. The second indication corresponds to the set of addresses including the destination address of the packet.